UC Davis
UC Davis Previously Published Works

Title

Interpretable Artificial Neural Networks incorporating Bayesian Alphabet Models for Genome-wide Prediction and Association Studies

Authors

Zhao, Tianjing
Fernando, Rohan
Cheng, Hao

Interpretable artificial neural networks incorporating Bayesian alphabet models for genome-wide prediction and association studies

Tianjing Zhao,[1,2] Rohan Fernando (ID) ,[3] and Hao Cheng (ID) [1,*]

[1]Department of Animal Science, University of California Davis, Davis, CA 95616, USA,
[2]Integrative Genetics and Genomics Graduate Group, University of California Davis, Davis, CA 95616, USA, and
[3]Department of Animal Science, Iowa State University, Ames, IA 50011, USA

*Corresponding author: Email: qtlcheng@ucdavis.edu

Abstract

In conventional linear models for whole-genome prediction and genome-wide association studies (GWAS), it is usually assumed that the relationship between genotypes and phenotypes is linear. Bayesian neural networks have been used to account for non-linearity such as complex genetic architectures. Here, we introduce a method named NN-Bayes, where "NN" stands for neural networks, and "Bayes" stands for Bayesian Alphabet models, including a collection of Bayesian regression models such as BayesA, BayesB, BayesC, and Bayesian LASSO. NN-Bayes incorporates Bayesian Alphabet models into non-linear neural networks via hidden layers between single-nucleotide polymorphisms (SNPs) and observed traits. Thus, NN-Bayes attempts to improve the performance of genome-wide prediction and GWAS by accommodating non-linear relationships between the hidden nodes and the observed trait, while maintaining genomic interpretability through the Bayesian regression models that connect the SNPs to the hidden nodes. For genomic interpretability, the posterior distribution of marker effects in NN-Bayes is inferred by Markov chain Monte Carlo approaches and used for inference of association through posterior inclusion probabilities and window posterior probability of association. In simulation studies with dominance and epistatic effects, performance of NN-Bayes was significantly better than conventional linear models for both GWAS and whole-genome prediction, and the differences on prediction accuracy were substantial in magnitude. In real-data analyses, for the soy dataset, NN-Bayes achieved significantly higher prediction accuracies than conventional linear models, and results from other four different species showed that NN-Bayes had similar prediction performance to linear models, which is potentially due to the small sample size. Our NN-Bayes is optimized for high-dimensional genomic data and implemented in an open-source package called "JWAS." NN-Bayes can lead to greater use of Bayesian neural networks to account for non-linear relationships due to its interpretability and computational performance.

Keywords: neural networks; Bayesian regression models; JWAS; genomic prediction; GWAS

Introduction

Genomic prediction plays an important role in animal and plant breeding by using genomic information to estimate genotypic values or breeding values of complex traits. The adoption of genomic information, such as single-nucleotide polymorphisms (SNPs), has greatly shortened the generation interval and improved the prediction accuracy (Meuwissen *et al.* 2001; Heffner *et al.* 2009; Hayes *et al.* 2009a; Hickey *et al.* 2017). Genome-wide association studies (GWAS) are used to detect associations between SNPs and traits. It has been applied in human diseases (*e.g.*, Ozaki *et al.* 2002; Hirschhorn and Daly 2005; Klein *et al.* 2005; Visscher *et al.* 2012, 2017; Buniello *et al.* 2019), as well as for traits of interest in animals and plants (*e.g.*, Atwell *et al.* 2010; Korte and Farlow 2013; Sharma *et al.* 2015; Freebern *et al.* 2020).

Based on the work of Fisher (1918), we typically assume that complex traits are affected by many genes with small additive effects, and that the relationship between genotypes and phenotypes is linear. Based on this assumption, the landmark paper of Meuwissen *et al.* (2001) introduced Bayesian regression models for whole-genome prediction. In Meuwissen *et al.* (2001), they proposed a model referred to as BayesA, where the marker effects were assigned a Student's-t prior distribution, and a model BayesB, where *a priori* only a proportion of markers had non-zero effects. Since Meuwissen *et al.* (2001), several variations to these regression models have been proposed, and, hereafter, they will be collectively referred as Bayesian Alphabet models. Most of the widely-used Bayesian regression models, including BayesA, BayesB, BayesC (Kizilkaya *et al.* 2010; Habier *et al.* 2011), Bayesian LASSO (Park and Casella 2008; Gianola and Fernando 2019), and BayesR (Erbe *et al.* 2012; Moser *et al.* 2015), differ only in the prior used for the marker effects. Another well-known linear model for genomic prediction is genomic best linear unbiased prediction

(GBLUP) (Habier *et al.* 2007; VanRaden 2008; Hayes *et al.* 2009b), where a genomic relationship matrix is used to account for the covariances among genetic values. It has been shown that GBLUP is equivalent to a Bayesian regression model with a normal prior for the marker effects (Fernando 1998; Habier *et al.* 2007; Strandén and Garrick 2009). These models, including their variations, have been widely used in GWAS (Wang *et al.* 2012, 2016; Moser *et al.* 2015; Fernando *et al.* 2017; Legarra *et al.* 2018). The assumption of linearity in these analyses, however, may affect their performance on genome-wide prediction and association studies (Nelson *et al.* 2013). Thus, machine learning models have been suggested due to their ability to capture the intricate non-linear relationship between high-dimensional inputs (genotypes) and outputs (phenotypes) (Szymczak *et al.* 2009; Azodi *et al.* 2019; Montesinos-López *et al.* 2021).

Artificial neural networks, inspired by information processing of brain, are a subset of machine learning models. Neural networks utilize multi-layer architectures to learn the representation of data (LeCun *et al.* 2015), where the processing layers are composed of neurons (*i.e.*, nodes) converting the input data into a more abstract representation. Neural networks have demonstrated their predictive ability in many fields such as speech recognition, object recognition, and detection (LeCun *et al.* 2015). It has also been applied in biological fields such as protein structure prediction, protein classification, and brain decoding (Min *et al.* 2017).

In some genomic prediction studies, neural networks with different architectures were compared to conventional linear models, such as GBLUP and Bayesian regression models. In genomic prediction applications, single hidden layer neural networks are typically applied, but they vary in the number of nodes in the hidden layer (Gianola *et al.* 2011; Okut *et al.* 2011, 2013; Ehret *et al.* 2015). Neural networks achieved higher prediction accuracies in some studies (*e.g.*, Gianola *et al.* 2011; Ma *et al.* 2018), but a gain in prediction performance was not consistently observed (*e.g.*, Okut *et al.* 2013; Bellot *et al.* 2018; Azodi *et al.* 2019; Abdollahi-Arpanahi *et al.* 2020), potentially due to the sample sizes in these studies or genetic architecture of the traits (Abdollahi-Arpanahi *et al.* 2020; Montesinos-López *et al.* 2021).

Besides the absence of a substantial improvement in prediction accuracy in genomic prediction, a major criticism of neural network models is the lack of interpretability. For example, deep learning ("deep" typically refers to neural networks with more than two hidden layers (Schmidhuber 2015)) is applied in a black-box manner, *i.e.*, it is hard to interpret results biologically for genome-wide analyses, although the neural network architecture may capture the complex relationships between inputs (genotypes) and outputs (phenotypes). Unlike neural networks, conventional linear models provide an explicit interpretation to link phenotypes to genotypes under some assumptions, and thus, they are used in many genome-wide analyses, *e.g.*, GWAS. This lack of interpretability of results from neural networks impedes their application for association studies (Ehret *et al.* 2015; Waldmann 2018; Montesinos-López *et al.* 2021). In some studies (*e.g.*, Okut *et al.* 2011; Glória *et al.* 2016; Waldmann 2018), weights in neural networks were interpreted as regression coefficients to obtain the effect of each SNP, and these effects without significance measures (*e.g.*, *P*-value) were used to rank all SNPs.

Bayesian regularization has been applied to the neural networks to reduce their complexity by treating the networks weights as random variables with prior densities (*e.g.*, Gianola *et al.* 2011; Okut *et al.* 2011, 2013; Glória *et al.* 2016; van Bergen *et al.* 2020). However, in most Bayesian neural network studies on

genomic prediction, in part due to computational reasons, inferences of unknowns were only based on the posterior mode via the maximum a posteriori (MAP) estimator (*e.g.*, Gianola *et al.* 2011; Okut *et al.* 2011, 2013; Glória *et al.* 2016). Chen *et al.* (2017) showed that in association studies, the performance of MAP inferences may be inferior to Markov chain Monte Carlo (MCMC) approaches, which exactly estimate the posterior distribution. In Demetci *et al.* (2021), variational inference was used to approximate the posterior distribution of posterior inclusion probabilities (PIPs) for all SNPs. However, simulations in Demetci *et al.* (2021) showed that such approximation may mis-estimate the network weights.

Although Bayesian methods had been proposed for analyses of genetic populations (Dempfle 1977; Gianola and Fernando 1986) in the 1980's, these methods were not widely adopted until MCMC approaches were introduced to draw inferences from posterior distributions (Wang *et al.* 1994). In the method proposed here, the posterior distribution of marker effects is inferred using MCMC approaches, and it is used to produce significance measures such as PIP (Guan and Stephens 2011) and window posterior probability of association (WPPA) (Fernando *et al.* 2017). The method is called NN-Bayes, where "NN" stands for neural networks, and "Bayes" stands for Bayesian Alphabet models, a collection of Bayesian regression models such as BayesA, BayesB, BayesC, and Bayesian LASSO. As shown in Figure 1, the framework of NN-Bayes starts with SNPs that are linearly connected to hidden nodes, through a multi-trait, multiple regression model, that are then non-linearly connected to the observed trait. Multi-trait Bayesian Alphabet models (Cheng *et al.* 2018b) will be employed to fit the regression model that connects the SNPs to the hidden nodes. The hidden nodes, sampled via Hamiltonian Monte Carlo, represent unobserved intermediate traits that are non-linearly connected to the observed trait to accommodate
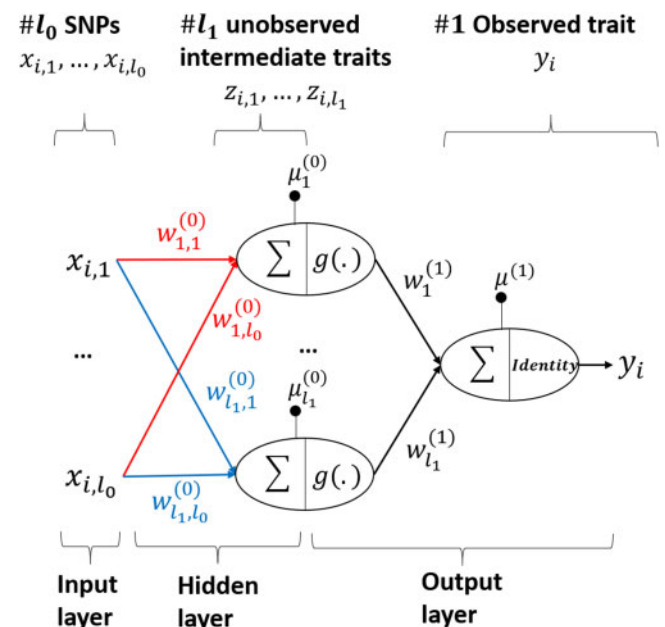

Figure 1 Framework of NN-Bayes. SNPs are linearly connected to unobserved intermediate traits (*i.e.*, hidden nodes). Multi-trait Bayesian Alphabet models will be employed to sample marker effects on hidden nodes (*i.e.*, weights between input and hidden layers). Then hidden nodes are non-linearly connected to the observed trait to accommodate complex genetic architecture including non-additive genetic effects, and the non-linearity is achieved by the non-linear function $g(.)$.
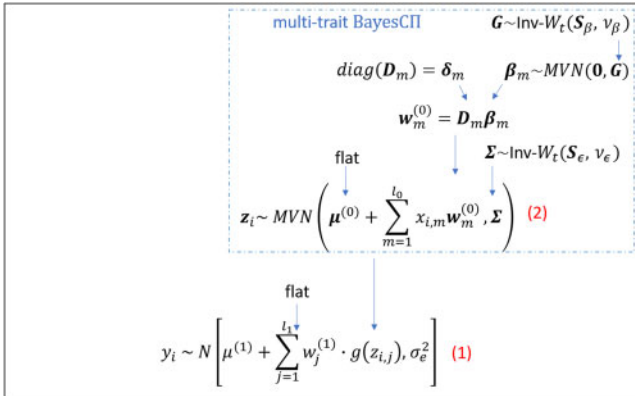
complex genetic architecture including non-additive genetic effects. In summary, NN-Bayes attempts to improve the performance of genome-wide prediction and GWAS by accommodating non-linear relationships between the hidden nodes and the observed trait, while maintaining genomic interpretability through the Bayesian regression models that connect the SNPs to the hidden nodes.

Compared to other neural network applications of genome-wide prediction and association studies, our NN-Bayes using MCMC approaches is optimized for high-dimensional genomic data to be computationally efficient. In most neural network applications of genomic prediction, neural networks were implemented using existing general purpose software tools which are not optimized for genomic analyses, and it would be time consuming to estimate effects of a large number of genome-wide molecular markers (*i.e.*, weights between input layer and hidden nodes). For example, in van Bergen *et al.* (2020), it took about 4 hours to run 1000 iterations for a single-layer neural network with 20 hidden nodes using a dataset of 500 individuals and 5000 SNPs. However, the computing time of our NN-Bayes, implemented in an open-source package "JWAS" (Cheng *et al.* 2018a), is about 2 minutes for such an analysis on a laptop (with 2.6 GHz Intel Core i7 processor), and it scales nearly linearly by the number of observations, the number of markers, and the number of hidden nodes. More details and examples are available in the JWAS documentation https://github.com/reworkhow/JWAS.jl.

Materials and methods
Bayesian neural network alphabet

Bayesian analyses using NN-Bayes will be demonstrated for a single observed trait. The hierarchical Bayesian model of NN-Bayes

Figure 2 NN-Bayes represented as a hierarchical Bayesian model. Mixture priors in multi-trait BayesCΠ are used for marker effects on unobserved intermediate traits.

is shown in Figure 2. For individual i, the observed trait is modeled as

$$y_i = \mu^{(1)} + \sum_{j=1}^{l_1} w_j^{(1)} g(z_{i,j}) + e_i, \qquad (1)$$

where y_i is the observed phenotype for individual i, $\mu^{(1)}$ is the overall mean, $z_{i,j}$ is the jth hidden node for individual i, $g(.)$ is an element-wise non-linear function, $w_j^{(1)}$ is the effect of $g(z_{i,j})$ on y_i, and e_i is a random residual. Non-linearity in the model is achieved through use of the function $g(.)$, which is called an activation function in the artificial neural network literature (Leshno *et al.* 1993), and the hyperbolic tangent function was applied in this paper. Flat priors are used for the weights $w_j^{(1)}$ and the overall mean $\mu^{(1)}$; the prior for e_i is a normal distribution with null mean and unknown variance σ_e^2, *i.e.*, $e_i \overset{i.i.d}{\sim} N(0, \sigma_e^2)$. A scaled inverse chi-squared distribution is assigned as the prior for σ_e^2, *i.e.*, $(\sigma_e^2 | \nu_e, S_e^2) \sim \nu_e S_e^2 \chi_{\nu_e}^{-2}$.

For ith individual, the prior for the hidden nodes, $\mathbf{z}_i = [z_{i,1}, \ldots, z_{i,l_1}]^T$, can be presented as a multi-trait Bayesian regression model (Cheng *et al.* 2018b):

$$\mathbf{z}_i = \boldsymbol{\mu}^{(0)} + \sum_{m=1}^{l_0} x_{i,m} \mathbf{w}_m^{(0)} + \boldsymbol{\epsilon}_i, \qquad (2)$$

where $\mathbf{z}_i$ is the vector of l_1 hidden nodes, which can be thought of as unobserved intermediate traits, for individual i, $\boldsymbol{\mu}^{(0)} = [\mu_1^{(0)} \cdots \mu_{l_1}^{(0)}]^T$ is a vector of overall means for the l_1 hidden nodes, $x_{i,m}$ is the observed genotype covariate at locus m for individual i (coded as 0,1,2), $\mathbf{w}_m^{(0)} = [w_{1,m}^{(0)} \cdots w_{l_1,m}^{(0)}]^T$ are the marker effects of locus m on the l_1 unobserved intermediate traits (*i.e.*, weights between m-th input node and all hidden nodes in the neural network), and $\boldsymbol{\epsilon}_i$ is a vector of random residuals for the l_1 hidden nodes.

The overall means, $\boldsymbol{\mu}^{(0)}$, are assigned flat priors. Conditional on $\boldsymbol{\Sigma}$, the residuals, $\boldsymbol{\epsilon}_i$, have independently and identically distributed multivariate normal priors with null means and covariance matrix $\boldsymbol{\Sigma}$, which itself is assumed to have an inverse Wishart prior distribution, $W_t^{-1}(\mathbf{S}_\epsilon, \nu_\epsilon)$. As shown in Cheng *et al.* (2018b), by employing the data augmentation, $\mathbf{w}_m^{(0)}$ can be written as $\mathbf{w}_m^{(0)} = \mathbf{D}_m \boldsymbol{\beta}_m$, where $\mathbf{D}_m$ is a diagonal matrix, where the kth diagonal entry indicates whether the marker effects of locus m for the unobserved intermediate trait k is zero or not. $\boldsymbol{\beta}_m$ follows some multivariate distribution, *e.g.*, the multivariate normal distribution in multi-trait BayesCΠ or the multivariate t distribution in multi-trait BayesB. Note that multi-trait Bayesian regression models may be computationally intensive if a large number of hidden nodes are used. In this case, we assume that hidden nodes are independent such that multiple single-trait Bayesian regression models can be used in parallel at each iteration.

In Figure 1, we also present the hierarchical Bayesian model described above as a single hidden layer feed-forward Bayesian neural network. This architecture is typical for neural networks used in genomic prediction (Gianola *et al.* 2011; Okut *et al.* 2011, 2013; Ehret *et al.* 2015).

In the neural network, for each observation, the input layer is composed of a vector of l_0 molecular markers (coded as 0, 1, 2), and the output layer is the observed trait. Nodes in the hidden layer, between input and output layer, represent unobserved intermediate traits, and the relationship between hidden nodes and observed trait is non-linear. The weights between the input layer and the hidden layer represent marker effects on

unobserved intermediate traits, the "biases" of hidden nodes can be regarded as the overall means. The weights and bias between the hidden layer and the output layer help to define the non-linear relationship between unobserved intermediate traits and observed trait. Note that extra input nodes can be included in the neural network if additional random or fixed effects are fitted in the model.

Inference using Markov chain Monte Carlo

From a Bayesian perspective, inferences of unknowns are based on their posterior distributions. However, closed-form expressions of those posteriors are usually unavailable. Thus, in practice, samples are obtained using MCMC techniques, where statistics computed from the resulting Markov chain converge to those from the posterior as chain length increases (Norris 1998; Sorensen and Gianola 2007). Here, to construct the Markov chain, we will use Gibbs sampling, where each unknown variable or block of variables is sampled from its full conditional distribution, conditioned on the observed data and the latest samples of all the other unknowns.

Given the sampled values of hidden nodes $\mathbf{z}_i$ in equation (2), unknown parameters between the input layer and the hidden layer do not depend on the observed trait. Sampling of those unknowns, including the overall means $\boldsymbol{\mu}^{(0)}$, the marker effects $\mathbf{w}_m^{(0)}$, and the residual covariance matrix $\boldsymbol{\Sigma}$, given the sampled values of $\mathbf{z}_i$, is based on the multi-trait Bayesian regression models described in Cheng *et al.* (2018b). Given the sampled values for the hidden nodes (after activation) $g(\mathbf{z}_i)$ and output node y_i, the sampling of weights and bias between the hidden layer and the output layer is straightforward (derivations are given in the Appendix). As shown below, we consider Hamiltonian Monte Carlo to draw samples for the hidden nodes. The sampler is implemented in the open-source package "JWAS" (Cheng *et al.* 2018a).

Sampling hidden nodes using Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) is used to sample hidden nodes, *i.e.,* $\mathbf{z}_i$. In HMC, each unknown parameter is paired with a "momentum" variable $\boldsymbol{\phi}_i$. The HMC constructs the Markov chain by a series of iterations. A useful introduction to the principles and concepts underlying HMC is given by Betancourt (2017). Following notation in Gelman *et al.* (2013), there are three steps in each iteration of the HMC:

1) updating the momentum variable independently of the current values of the paired parameter, *i.e.,* $\boldsymbol{\phi}_i \sim MVN(\mathbf{0}, \mathbf{M})$.
2) updating $(\mathbf{z}_i, \boldsymbol{\phi}_i)$ via L "leapfrog steps." In each leapfrog step, $\mathbf{z}_i$ and $\boldsymbol{\phi}_i$ are updated dependently and scaled by ϵ. The leapfrog step below is repeated L times:

    a) $\boldsymbol{\phi}_i \leftarrow \boldsymbol{\phi}_i + \frac{1}{2}\epsilon \frac{dlogp(\mathbf{z}_i|y_i, ELSE)}{d\mathbf{z}_i}$;
    b) $\mathbf{z}_i \leftarrow \mathbf{z}_i + \epsilon \mathbf{M}^{-1} \boldsymbol{\phi}_i$;
    c) $\boldsymbol{\phi}_i \leftarrow \boldsymbol{\phi}_i + \frac{1}{2}\epsilon \frac{dlogp(\mathbf{z}_i|y_i, ELSE)}{d\mathbf{z}_i}$

The resulting state at the end of L repetitions will be denoted as $(\mathbf{z}_i^*, \boldsymbol{\phi}_i^*)$.

3) calculating the acceptance rate, r, and above resulting state will be accepted with probability $min(1, r)$.

In our method, the gradient of log full conditional posterior distribution of $\mathbf{z}_i$ is:

$$\frac{dlogf(\mathbf{z}_i|y_i, ELSE)}{d\mathbf{z}_i}$$
$$= -\boldsymbol{\Sigma}^{-1}(\mathbf{z}_i - \boldsymbol{\mu}^{(0)} - \sum_{m=1}^{l_0} x_{i,m}\mathbf{w}_m^{(0)}) \qquad (3)$$
$$+ \frac{y_i - \mu^{(1)} - (\mathbf{w}^{(1)})^T g(\mathbf{z}_i)}{\sigma_e^2}\mathbf{w}^{(1)} \circ g'(\mathbf{z}_i),$$

where $\mathbf{w}^{(1)} = [w_1^{(1)}, \ldots, w_{l_1}^{(1)}]^T$ is the vector of weights between hidden nodes and observed trait, *ELSE* denotes the sampled values of remaining unknowns, and the symbol "$\circ$" denotes the element-wise production.

In the round t of HMC, the acceptance rate, r is:

$$r = e^{log(p(\mathbf{z}_i^*|y_i, ELSE)p(\boldsymbol{\phi}_i^*)) - log(p(\mathbf{z}_i^{[t-1]}|y_i, ELSE)p(\boldsymbol{\phi}_i^{[t-1]}))} \qquad (4)$$

A detailed derivation can be found in the Appendix. In our analyses, 10 leapfrog steps was applied in each iteration of HMC, *i.e.,* $L = 10$, the scale parameter ϵ was 0.1, and $\mathbf{M}$ was set as an identity matrix.

NN-Bayes for genomic prediction

The MCMC sample from the posterior distribution of genotypic values for individuals of interest is obtained as

$$\hat{\mathbf{g}} = g(\mathbf{1}\boldsymbol{\mu}^{(0)\prime} + \mathbf{X}\mathbf{w}^{(0)})\mathbf{w}^{(1)}, \qquad (5)$$

where $\mathbf{X}$ is the genotype covariate matrix for individuals of interest, $\mathbf{w}^{(0)}$ is a matrix of size p-by-l_1 of samples of marker effects on the l_1 hidden nodes, $\boldsymbol{\mu}^{(0)}$ is a vector of samples of overall means for the l_1 hidden nodes, and $\mathbf{w}^{(1)}$ is a vector of samples of l_1 weights between hidden nodes and the observed trait, and $g(.)$ is a non-linear function. The prediction accuracy is calculated as the Pearson correlation between the posterior mean of genotypic values and the phenotypic values in the validation dataset.

NN-Bayes for genome-wide association studies

In NN-Bayes, the PIP for each single marker is computed as the frequency that its effect is non-zero on at least one of hidden nodes. Here we prefer the use of the posterior probability of association of a genomic window (WPPA) (Fernando *et al.* 2017) to account for the fact that highly correlated SNPs within a genomic window jointly affect the phenotype, and it is difficult to identify the effect of a single marker (Hayes *et al.* 2010).

Given that an investigator is interested in identifying genomic segments that explain more than a proportion T of the genetic variance, WPPA is defined as the posterior probability of this event (Fernando *et al.* 2017). To estimate WPPA, first, MCMC samples of genotypic values for all individuals are obtained from their posterior distribution, using equation (5). Next, using the notation in equation (5), the genotypic values that are attributed to genomic window t are sampled from their posterior distribution as

$$\hat{\mathbf{g}}_t = g(\mathbf{X}_t\mathbf{w}_t^{(0)})\mathbf{w}^{(1)}, \qquad (6)$$

where $\mathbf{X}_t$ is the genotype covariate matrix of markers in window t, and $\mathbf{w}_t^{(0)}$ represents the samples of marker effects in window t on hidden nodes. The proportion of the genetic variance explained by the genomic window t, q_t, is now sampled as $\frac{\sigma_{a_t}^2}{\sigma_a^2}$, where σ_a^2 and $\sigma_{a_t}^2$ are the variances of a random sample from $\mathbf{g}$ and $\mathbf{g}_t$, respectively. Then, the posterior probability that window t accounts for more than a proportion T of the genetic variance (*i.e.,* WPPA) can be estimated from the MCMC samples by

counting the number of samples where $q_t > T$ (Fernando and Garrick 2013; Chen et al. 2017; Lloyd-Jones et al. 2017). In this article, non-overlapping windows of 1 Mb were used to identify genomic windows that explain over 1% of the total genetic variance (i.e., $T = 0.01$)

Data analysis
Simulated data

Real pig genotypes in Duarte et al. (2014) were used to simulate phenotypes with dominance and epistatic effects. All SNP markers on the chromosome 1 were used, resulting a genotypic data with 5023 markers for 928 individuals. A random sample of 5% of these 5023 markers were selected as quantitative trait loci (QTL). Following the simulation in van Bergen et al. (2020), the additive effects (a) of QTL were sampled from a univariate normal distribution with null mean and variance one. Dominance factors (δ) were sampled from a univariate normal distribution with mean 1.2 and standard deviation 0.3. Dominance effects (d) were computed as $\delta|a|$. The epistatic factors (γ) of all pairwise combinations of QTL were sampled from a univariate normal distribution with null mean and variance one. The epistatic effect (ϵ) between ith and jth QTL was computed as $\gamma\sqrt{|a_i a_j|}$. The complementary epistasis scenario in van Bergen et al. (2020) was used to simulate the total genetic value. Phenotypes were simulated with a broad-sense heritability of 0.5 and were scaled to have a phenotypic variance equal to one. Based on the simulation processes described above, 20 different datasets were simulated.

Real data

Publicly available genotypic and phenotypic data of multiple species were used to compare NN-Bayes and conventional linear models. These datasets include the pig dataset from Duarte et al. (2014), and data for soy, maize, switchgrass, and spruce from Azodi et al. (2019). Traits used in our analyses were the 13 week tenth rib backfat (BF), yield (YLD), wood density (DE) and flowering time (FT). The description of each dataset, such as the number of SNP markers and the number of observations, are showed in Table 1.

Genomic prediction

Twenty and fifty replicates were applied for each real and simulated dataset, respectively. In each replicate, a random subset of 80% of all observations were used for training, and the remaining were used for validation. The prediction accuracy is calculated as the Pearson correlation between the posterior mean of genotypic values and the phenotypic values in the validation dataset. In NN-Bayes, different priors for marker effects (i.e., weights between inputs and hidden nodes) in Bayesian Alphabet models

(i.e., RR-BLUP, BayesA, BayesB, BayesCπ, and Bayesian LASSO) were used. Linear models with priors in Bayesian Alphabet models were also performed. Different number of hidden nodes (i.e., 2, 3, 5 and 10) were tested in the analyses. Chains of length 10,000 and 20,000 were applied to simulated datasets and real datasets, respectively, to ensure the convergence.

Genome-wide association studies

For each dataset, PIP and WPPA were used for association inference. In the simulated study, non-overlapping windows of 1 Mb were tested. Genomic windows that explain over 1% of the total genetic variance were assumed to be of potential interest (i.e., $T = 0.01$). The area under the receiver operating characteristic curve (AUC) was calculated using the R package pROC (Robin et al. 2011) to assess the performance of NN-Bayes in GWAS.

Results
Genomic prediction
Simulated data

Overall, the prediction accuracies of NN-Bayes were significantly higher than those of conventional linear models, and the differences were substantial in magnitude. The number of hidden nodes did not significantly affect the performance of NN-Bayes. No significant differences were found with different priors for marker effects used in NN-Bayes. Here, we only present the comparison of prediction accuracy between NN-Bayes composed of 10 hidden nodes with BayesCΠ priors (named as NN-BayesCΠ-10) and the linear model with BayesCπ prior (LM-BayesCπ). In Figure 3, the 20 simulated datasets were distinguished by color. Prediction accuracies of NN-BayesCΠ-10 were higher than LM-BayesCπ in 949 out of 1,000 validation sets (i.e., above the diagonal black line), and the prediction accuracy for NN-BayesCΠ-10 was significantly higher than for LM-BayesCπ in 16 out of 20 simulated datasets under the t-test with a significance level of 0.05. The results show that our NN-Bayes has the potential to improve the prediction accuracy for a trait with non-additive genetic effects.

Real data

In each of the five species, the prediction accuracies of NN-Bayes with different numbers of hidden nodes were not significantly different under the t-test with a significance level of 0.05. Significant differences were found among different priors for marker effects in the pig data. Comparison of prediction accuracies among all methods is shown in Table 2 in the Appendix. In detail, NN-Bayes using BayesA and BayesB priors performed better than methods using other priors in the pig data. The

Table 1 Comparison among five species in prediction accuracies of NN-Bayes composed of 10 hidden nodes with BayesCΠ prior (NN-BayesCΠ-10) and a linear model with BayesCπ prior (LM-BayesCπ)

					Prediction accuracy	
Reference	Species	Trait[a]	#SNPs	#Observations	NN-BayesCΠ-10	LM-BayesCπ
Duarte et al. (2014)	Pig	BF	42,246	928	0.530 (0.05)	0.533 (0.04)
Azodi et al. (2019)	Soy	YLD	5014	4234	0.663 (0.02)	0.656 (0.02)
Azodi et al. (2019)	Spruce	DE	6785	1722	0.418 (0.05)	0.412 (0.05)
Azodi et al. (2019)	Switchgrass	FT	217,150	514	0.860 (0.03)	0.865 (0.03)
Azodi et al. (2019)	Maize	FT	244,771	391	0.754 (0.10)	0.757 (0.09)

[a] BF: 13 week tenth rib backfat; YLD: yield; DE: wood density; FT: flowering time; #SNP, number of SNP markers; #Observations, number of observations; Prediction accuracy: the average prediction accuracy from 20 replicates with the standard deviation in the bracket.
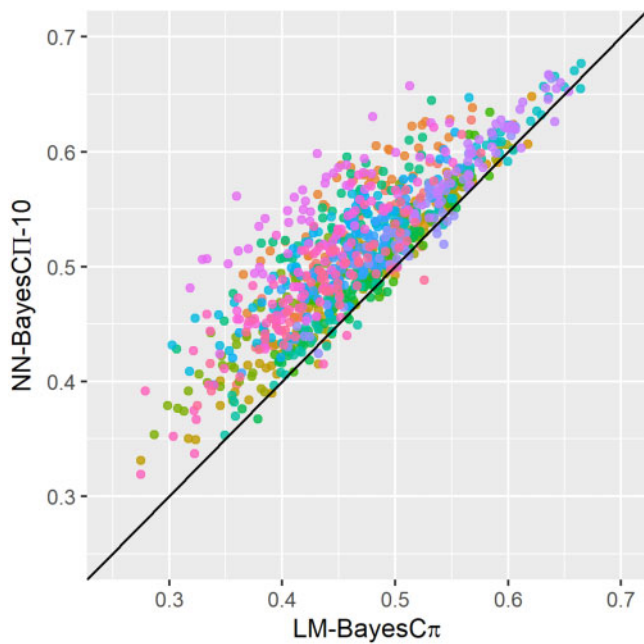
Figure 3 The prediction accuracy of NN-Bayes composed of 10 hidden nodes with BayesCΠ priors (NN-BayesCΠ-10) versus a linear model with BayesCπ prior (LM-BayesCπ) on scenarios with both additive and non-additive effects (*i.e.*, dominance and epistasis). Twenty simulated datasets were distinguished by color. For each dataset, 50 replicates for validation were applied. The diagonal line is used for reference such that a dot above the line represents a validation with higher accuracy for NN-Bayes.

prediction accuracies of NN-Bayes composed of 10 hidden nodes with BayesCΠ priors (named as NN-BayesCΠ-10) and the linear model with BayesCπ prior (LM-BayesCπ) are showed in the last column of Table 1. Among the five species, the prediction accuracies for NN-BayesCΠ-10 were higher than those for LM-BayesCπ when the ratio between number of observations and number of markers was relatively high (*e.g.*, soy and spruce). However, the prediction accuracies of NN-BayesCΠ-10 were not significantly different from those for LM-BayesCπ under the t-test with a significance level of 0.05. This may be because the sample size is not large enough for accurate parameter estimates, or the architecture of the neural network may not be sufficiently complex to represent the intricate hidden relationships between genotypes and phenotype. Alternatively, the additive genetic effects may have already accounted for a majority proportion of the variation, so the improvement based on non-additive effects is limited. When we considered a significance level of 0.1, the prediction accuracies for NN-Bayes composed of 10 hidden nodes with priors for marker effects in BayesCΠ or RR-BLUP were significantly higher than conventional linear models for the soy dataset. As showed in Figure 4, for the soy dataset, in 15 out of 20 validation sets, the NN-BayesCΠ-10 had a higher prediction accuracy than the LM-BayesCπ (*i.e.*, above the diagonal black line).

Genome-wide association studies
Simulated data
GWAS was conducted on 20 simulated datasets. NN-Bayes composed of 2 hidden nodes with priors for marker effects in BayesCΠ (named as NN-BayesCΠ-2) was tested and compared to results from a linear model with BayesCπ prior (LM-BayesCπ). Chain of length 500,000 with 200,000 burn-in was used to ensure the convergence, and samples were saved for every 50 iterations.

For both NN-BayesCΠ-2 and LM-BayesCπ, when PIP was used for association inference, the AUC was around 0.5, indicating the performance of PIP was similar to a random classifier. Thus, we will only present the GWAS results using WPPA in all analyses. The AUC of NN-BayesCΠ-2 was significantly higher than that of LM-BayesCπ, under the t-test with a significance level of 0.15. As showed in Figure 5, the AUC for NN-BayesCΠ-2 was higher than that for LM-BayesCπ in 16 out of the 20 simulated datasets (*i.e.*, above the diagonal black line).

Real data
GWAS was conducted on the real pig dataset. NN-Bayes composed of two hidden nodes with priors for marker effects in BayesCΠ (named as NN-BayesCΠ-2) was tested. Chain of length 1 million with 500,000 burn-in was used to ensure the convergence, and samples were saved for every 10 iterations.

For NN-BayesCΠ-2, a WPPA >0.95 was used to identify significant associations because it results in controlling the proportion of false positives to ≤ 0.05 (Fernando *et al.* 2017). The GWAS results from NN-BayesCΠ-2 had a similar pattern as those in Duarte *et al.* (2014). The significant window on chromosome 6 was identified, which is consistent with the GWAS results in Duarte *et al.* (2014).

Discussion

In conventional linear models used for genomic prediction, it is usually assumed that complex traits are affected by many genes with small additive effects, and that the relationship between genotypes and phenotypes is linear. To account for non-linear relationships such as non-additive genetic effects, we proposed a method named NN-Bayes, where "NN" stands for neural networks, and "Bayes" stands for Bayesian Alphabet models. NN-Bayes incorporates Bayesian Alphabet models into non-linear neural networks via hidden layers between SNPs and observed traits. Priors in multi-trait Bayesian Alphabet models are assumed for marker effects on hidden nodes, and flexible non-linear relationships between hidden nodes and the observed trait are assumed through activation functions, *e.g.*, the hyperbolic tangent function. Thus, NN-Bayes attempts to improve the accuracy of prediction by accommodating non-linear relationships between the hidden nodes and the observed trait, while maintaining genomic interpretability through the Bayesian regression methods that connect the SNPs to the hidden nodes. Compared to other neural network applications on genomic prediction, NN-Bayes has been optimized for MCMC based inference with high-dimensional genomic data, and it is thus more computationally efficient. Our NN-Bayes is implemented in an open-source package called "JWAS" (Cheng *et al.* 2018a).

In the analyses of the simulated data, where the phenotypes were simulated with both additive and non-additive effects (*i.e.*, dominance and epistasis), performance of NN-Bayes was significantly better than conventional linear models for both GWAS and whole-genome prediction, and the differences on prediction accuracy were substantial in magnitude. In real-data analyses, for the soy dataset, NN-Bayes was able to achieve significantly higher prediction accuracies than conventional linear models, and results from other four different species showed that NN-Bayes had similar prediction performance to linear models. Results in Azodi *et al.* (2019) showed that artificial neural networks did not tend to outperform linear models for the multiple plant datasets used in our analyses. This may be because the sample size is not large enough for accurate parameter
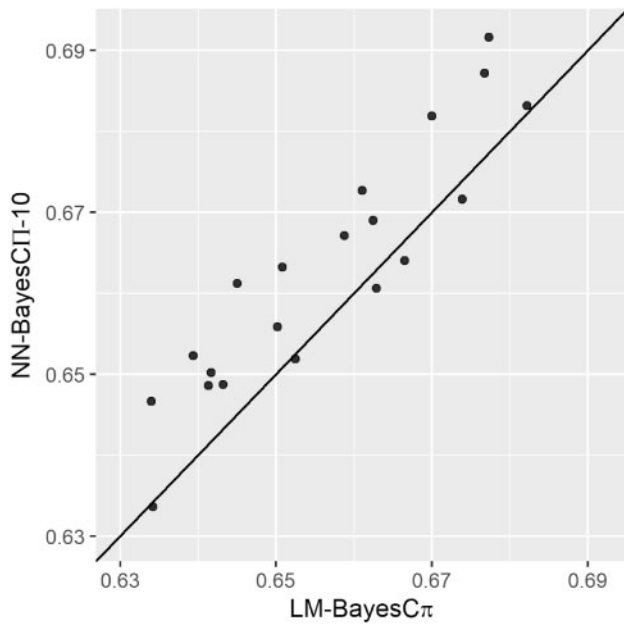
Figure 4 The prediction accuracy of NN-Bayes composed of 10 hidden nodes with BayesCΠ priors (NN-BayesCΠ-10) versus the linear model with BayesCπ prior (LM-BayesCπ) for the soy dataset. 20 replicates for validation were applied. The diagonal line is used for reference such that a dot above the line represents a validation with higher accuracy for NN-Bayes.
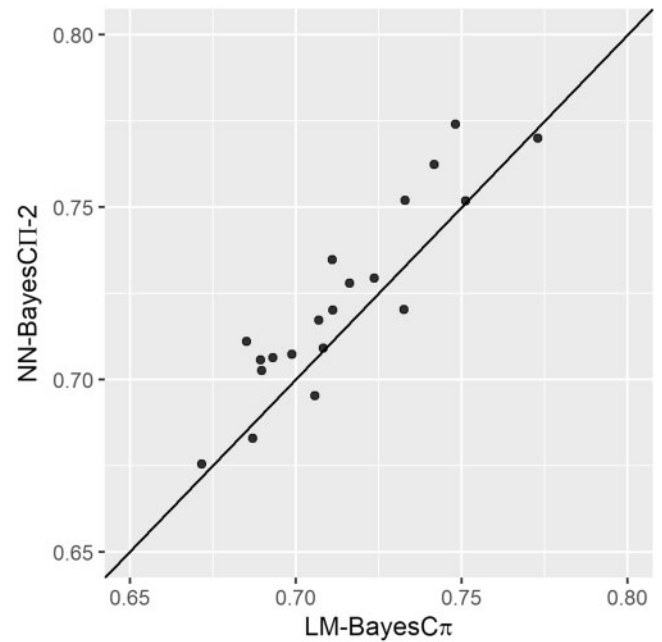

Figure 5 The AUC of GWAS results on 20 simulated datasets for NN-Bayes composed of two hidden nodes with BayesCΠ priors (NN-BayesCΠ-2) versus a linear model with BayesCπ prior (LM-BayesCπ), on scenarios with both additive and non-additive effects (*i.e.*, dominance and epistasis). Inference of association is based on genomic windows. The diagonal line is used for reference such that a dot above the line represents a simulated dateset with higher AUC for NN-Bayes.

estimates, or the architecture of the neural network may not be sufficiently complex to represent the intricate hidden relationships between genotypes and phenotype. Alternatively, the additive genetic effects may have already accounted for a majority proportion of the variation, so the improvement based on non-additive effects is limited.

In most neural network applications on genomic prediction, single hidden layer, feed-forward neural networks have been used (*e.g.*, Gianola *et al.* 2011; Okut *et al.* 2011, 2013; Ehret *et al.* 2015). These neural networks are often trained by gradient-based back-propagation, and the training process stops when it reaches the maximum number of training iterations or stops early when a criterion is met, for example, the optimal mean squared error on a validation dataset. To reduce the overfitting in such neural networks with large numbers of molecular markers as input, regularization technology, such as Bayesian regularization (Gianola *et al.* 2011) and L_1-norm penalty on unknown parameters (Wang *et al.* 2018), are usually applied. In Bayesian regularized neural networks, the effective number of parameters were similar when different numbers of neurons were used in the middle layer (Gianola *et al.* 2011; Okut *et al.* 2013; Ehret *et al.* 2015) and were usually much smaller than the sample size. Moreover, dropout may be applied to reduce the complexity of neural networks by setting a random proportion of weights to zero in each training iteration (Waldmann 2018; Azodi *et al.* 2019). In van Bergen *et al.* (2020), variable selection was used in a Bayesian neural network by fitting a "marker selection vector" into the Bayesian neural networks.

Most Bayesian neural network studies on genomic prediction heavily rely on approximations in part because they were performed using general algorithms implemented in existing software tools that were not optimized for genomic data analyses. For example, in Gianola *et al.* (2011) and Okut *et al.* (2013), the rationale of using relationship matrix as input of neural networks

is from the representation of the infinitesimal model as a regression on relationship matrix, so weights connecting the input and hidden layers should follow the prior distribution of $N(\mathbf{0}, \mathbf{A}^{-1}\sigma_w^2)$, where $\mathbf{A}$ is the relationship matrix. However, due to the use of MATLAB (Beale *et al.* 2010) in Gianola *et al.* (2011) and Okut *et al.* (2013), this prior was restricted in the form of $N(\mathbf{0}, \mathbf{I}\sigma_w^2)$. In addition to approximations, it is usually computationally intensive or infeasible to analyze large datasets on genomic prediction using general algorithms implemented in existing software tools. Okut *et al.* (2013) reported that MATLAB required about 2 h to train a neural network with ∼2500 SNPs as input, and van Bergen *et al.* (2020) reported that PyMC3 (Salvatier *et al.* 2016) took about 4 h with 500 individuals and 5000 SNPs. For a dense genotype, such as the Jersey cow data with 35,798 SNPs, the relationship matrix of size 297 was used as input to make computations feasible in MATLAB (Gianola *et al.* 2011). These methods (Gianola *et al.* 2011; Okut *et al.* 2011, 2013; Ehret *et al.* 2015; Bellot *et al.* 2018; Azodi *et al.* 2019) were not used for GWAS due to the lack of interpretability. Our NN-Bayes using MCMC approaches is computationally efficient and can be further sped up by a recently developed parallel computing strategy (Zhao *et al.* 2020).

Data availability

The genotypic and phenotypic data used in the real-data analyses are publicly available in Duarte *et al.* (2014) and Azodi *et al.* (2019). The simulated data and scripts are available at https:// github.com/zhaotianjing/NN-Bayes. The authors state that all data necessary for confirming the conclusions presented in the article are represented fully within the article.

Funding

Conflicts of interest

None declared.

Literature cited

Abdollahi-Arpanahi R, Gianola D, Peñagaricano F. 2020. Deep learning versus parametric and ensemble methods for genomic prediction of complex phenotypes. Genet Sel Evol. 52:15.

Atwell S, Huang YS, Vilhjálmsson BJ, Willems G, Horton M, et al. 2010. Genome-wide association study of 107 phenotypes in Arabidopsis thaliana inbred lines. Nature. 465:627–631.

Azodi CB, Bolger E, McCarren A, Roantree M, de Los Campos G, et al. 2019. Benchmarking parametric and machine learning models for genomic prediction of complex traits. G3 (Bethesda). 9: 3691–3702.

Bellot P, de Los Campos G, Pérez-Enciso M. 2018. Can deep learning improve genomic prediction of complex human traits? Genetics. 210:809–819.

Betancourt M. 2017. A conceptual introduction to hamiltonian monte carlo. arXiv. preprint arXiv:1701.02434.

Buniello A, MacArthur JAL, Cerezo M, Harris LW, Hayhurst J, et al. 2019. The NHGRI-EBI GWAS catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. Nucleic Acids Res. 47:D1005–D1012.

Chen C, Steibel JP, Tempelman RJ. 2017. Genome-wide association analyses based on broadly different specifications for prior distributions, genomic windows, and estimation methods. Genetics. 206:1791–1806.

Cheng H, Fernando R, and Garrick D. 2018a. JWAS: Julia implementation of whole-genome analysis software. In: Proceedings of the world congress on genetics applied to livestock production, Vol. 11. p. 859.

Cheng H, Kizilkaya K, Zeng J, Garrick D, Fernando R. 2018b. Genomic prediction from multiple-trait Bayesian regression methods using mixture priors. Genetics. 209:89–103.

Demetci P, Cheng W, Darnell G, Zhou X, Ramachandran S, et al. 2021. Multi-scale Inference of Genetic Trait Architecture using Biologically Annotated Neural Networks. bioRxiv, pp.2020–07.

Dempfle L. 1977. Relation entre BLUP (Best Linear Unbiased Prediction) et estimateurs Bayésiens. Genet Sel Evol. 9:27.

Beale M.H., Hagan, M.T. and Demuth, H.B., 2010. Neural network toolbox. User's Guide, MathWorks, 2, pp. 77–81.

Duarte JLG, Cantet RJ, Bates RO, Ernst CW, Raney NE, et al. 2014. Rapid screening for phenotype-genotype associations by linear transformations of genomic evaluations. BMC Bioinformatics. 15:246.

Ehret A, Hochstuhl D, Gianola D, Thaller G. 2015. Application of neural networks with back-propagation to genome-enabled prediction of complex traits in Holstein-Friesian and German Fleckvieh cattle. Genet Sel Evol. 47:22.

Erbe M, Hayes B, Matukumalli L, Goswami S, Bowman P, et al. 2012. Improving accuracy of genomic predictions within and between dairy cattle breeds with imputed high-density single nucleotide polymorphism panels. J Dairy Sci. 95:4114–4129.

Fernando R. 1998. Genetic evaluation and selection using genotypic, phenotypic and pedigree information. In Proceedings of the 6th

World Congress on Genetics Applied to Livestock Production, Vol. 26. Armidale, Australia. p. 329–336.

Fernando R, Toosi A, Wolc A, Garrick D, Dekkers J. 2017. Application of whole-genome prediction methods for genome-wide association studies: a Bayesian approach. JABES. 22:172–193.

Fernando RL, Garrick D. 2013. Bayesian methods applied to GWAS. In: Cedric Gondro et al. (eds.), Genome-Wide Association Studies and Genomic Prediction, Methods in Molecular Biology. Totowa, NJ: Humana Press. p. 237–274.

Fisher RA. 1918. Xv.—the correlation between relatives on the supposition of Mendelian inheritance. Trans R Soc Edinb. 52:399–433.

Freebern E, Santos DJ, Fang L, Jiang J, Gaddis KLP, et al. 2020. GWAS and fine-mapping of livability and six disease traits in Holstein cattle. BMC Genomics. 21:11.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, et al. 2013. Bayesian Data Analysis. CRC Press.

Gianola D, Fernando RL. 1986. Bayesian methods in animal breeding. J Anim Sci. 63:217–244.

Gianola D, Fernando RL. 2019. A multiple-trait Bayesian Lasso for genome-enabled analysis and prediction of complex traits. Genetics. 214:305–331.

Gianola D, Okut H, Weigel KA, Rosa GJ. 2011. Predicting complex quantitative traits with Bayesian neural networks: a case study with jersey cows and wheat. BMC Genet. 12:87.

Glória LS, Cruz CD, Vieira RAM, de Resende MDV, Lopes PS, et al. 2016. Accessing marker effects and heritability estimates from genome prediction by Bayesian regularized neural networks. Livestock Sci. 191:91–96.

Guan Y, Stephens M. 2011. Bayesian variable selection regression for genome-wide association studies and other large-scale problems. Ann Appl Stat. 5:1780–1815.

Habier D, Fernando RL, Dekkers JC. 2007. The impact of genetic relationship information on genome-assisted breeding values. Genetics. 177:2389–2397.

Habier D, Fernando RL, Kizilkaya K, Garrick DJ. 2011. Extension of the Bayesian alphabet for genomic selection. BMC Bioinformatics. 12:186.

Hayes BJ, Bowman PJ, Chamberlain AJ, Goddard ME. 2009a. Invited review: Genomic selection in dairy cattle: progress and challenges. J Dairy Sci. 92:433–443.

Hayes BJ, Pryce J, Chamberlain AJ, Bowman PJ, Goddard ME. 2010. Genetic architecture of complex traits and accuracy of genomic prediction: coat colour, milk-fat percentage, and type in Holstein cattle as contrasting model traits. PLoS Genet. 6:e1001139.

Hayes BJ, Visscher PM, Goddard ME. 2009b. Increased accuracy of artificial selection by using the realized relationship matrix. Genet Res (Camb). 91:47–60.

Heffner EL, Sorrells ME, Jannink J-L. 2009. Genomic selection for crop improvement. Crop Sci. 49:1–12.

Hickey JM, Chiurugwi T, Mackay I, Powell W, Eggen A, et al., Implementing Genomic Selection in CGIAR Breeding Programs Workshop Participants. 2017. Genomic prediction unifies animal and plant breeding programs to form platforms for biological discovery. Nat Genet. 49:1297–1303.

Hirschhorn JN, Daly MJ. 2005. Genome-wide association studies for common diseases and complex traits. Nat Rev Genet. 6:95–108.

Kizilkaya K, Fernando RL, Garrick DJ. 2010. Genomic prediction of simulated multibreed and purebred performance using observed fifty thousand single nucleotide polymorphism genotypes1. J Anim Sci. 88:544–551.

Klein RJ, Zeiss C, Chew EY, Tsai J-Y, Sackler RS, et al. 2005. Complement factor H polymorphism in age-related macular degeneration. Science. 308:385–389.

Korte A, Farlow A. 2013. The advantages and limitations of trait analysis with GWAS: a review. Plant Methods. 9:29.

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

Legarra A, Ricard A, Varona L. 2018. GWAS by GBLUP: single and multimarker EMMAX and Bayes factors, with an example in detection of a major gene for horse gait. G3 (Bethesda). 8:2301–2308.

Leshno M, Lin VY, Pinkus A, Schocken S. 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw. 6:861–867.

Lloyd-Jones LR, Robinson MR, Moser G, Zeng J, Beleza S, *et al.* 2017. Inference on the genetic Basis of eye and skin color in an admixed population via Bayesian linear mixed models. Genetics. 206: 1113–1126.

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

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

Min S, Lee B, Yoon S. 2017. Deep learning in bioinformatics. Brief Bioinform. 18:851–869.

Montesinos-López OA, Montesinos-López A, Pérez-Rodríguez P, Barrón-López JA, Martini JW, *et al.* 2021. A review of deep learning applications for genomic selection. BMC Genomics. 22:19.

Moser G, Lee SH, Hayes BJ, Goddard ME, Wray NR, *et al.* 2015. Simultaneous discovery, estimation and prediction analysis of complex traits using a Bayesian mixture model. PLoS Genet. 11: e1004969.

Nelson RM, Pettersson ME, Carlborg Ö. 2013. A century after fisher: time for a new paradigm in quantitative genetics. Trends Genet. 29:669–676.

Norris JR. 1998. Markov Chains. Number 2, Cambridge University Press.

Okut H, Gianola D, Rosa GJ, Weigel KA. 2011. Prediction of body mass index in mice using dense molecular markers and a regularized neural network. Genet Res (Camb). 93:189–201.

Okut H, Wu X-L, Rosa GJ, Bauck S, Woodward BW, *et al.* 2013. Predicting expected progeny difference for marbling score in Angus cattle using artificial neural networks and Bayesian regression models. Genet Sel Evol. 45:13.

Ozaki K, Ohnishi Y, Iida A, Sekine A, Yamada R, *et al.* 2002. Functional SNPs in the lymphotoxin-α gene that are associated with susceptibility to myocardial infarction. Nat Genet. 32:650–654.

Park T, Casella G. 2008. The Bayesian lasso. J Am Stat Assoc. 103: 681–686.

Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, *et al.* 2011. pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics. 12:8.

Salvatier J, Wiecki TV, Fonnesbeck C. 2016. Probabilistic programming in python using pymc3. PeerJ Comput Sci. 2:e55.

Schmidhuber J. 2015. Deep learning in neural networks: an overview. Neural Netw. 61:85–117.

Sharma A, Lee JS, Dang CG, Sudrajad P, Kim HC, *et al.* 2015. Stories and challenges of genome wide association studies in livestock—a review. Asian-Australas J Anim Sci. 28:1371–1379.

Sorensen D, Gianola D. 2007. Likelihood, Bayesian, and MCMC Methods in Quantitative Genetics. Springer Science & Business Media.

Strandén I, Garrick DJ. 2009. Technical note: derivation of equivalent computing algorithms for genomic predictions and reliabilities of animal merit. J Dairy Sci. 92:2971–2975.

Szymczak S, Biernacka JM, Cordell HJ, González-Recio O, König IR, *et al.* 2009. Machine learning in genome-wide association studies. Genet Epidemiol. 33:S51–S57.

van Bergen GH, Duenk P, Albers CA, Bijma P, Calus MP, *et al.* 2020. Bayesian neural networks with variable selection for prediction of genotypic values. Genet Sel Evol. 52:14.

VanRaden PM. 2008. Efficient methods to compute genomic predictions. J Dairy Sci. 91:4414–4423.

Visscher PM, Brown MA, McCarthy MI, Yang J. 2012. Five years of GWAS discovery. Am J Hum Genet. 90:7–24.

Visscher PM, Wray NR, Zhang Q, Sklar P, McCarthy MI, *et al.* 2017. 10 years of GWAS discovery: biology, function, and translation. Am J Hum Genet. 101:5–22.

Waldmann P. 2018. Approximate Bayesian neural networks in genomic prediction. Genet Sel Evol. 50:70.

Wang CS, Rutledge JJ, Gianola D. 1994. Bayesian analysis of mixed linear models via Gibbs sampling with an application to litter size in Iberian pigs. Genet Sel Evol. 26:91–115.

Wang H, Misztal I, Aguilar I, Legarra A, Muir W. 2012. Genome-wide association mapping including phenotypes from relatives without genotypes. Genet Res (Camb). 94:73–83.

Wang T, Chen Y-PP, Bowman PJ, Goddard ME, Hayes BJ. 2016. A hybrid expectation maximisation and MCMC sampling algorithm to implement Bayesian mixture model based genomic prediction and QTL mapping. BMC Genomics. 17:21.

Wang Y, Mi X, Rosa GJ, Chen Z, Lin P, *et al.* 2018. An R package for fitting sparse neural networks with application in animal breeding. J Anim Sci. 96:2016–2026.

Zhao T, Fernando R, Garrick D, Cheng H. 2020. Fast parallelized sampling of Bayesian regression models for whole-genome prediction. Genet Sel Evol. 52:11.

Appendix
MCMC in NN-Bayes

Sampling weights between hidden layers and output layers

The full conditional posterior distribution of $\mu^{(1)}$ and $\mathbf{w}^{(1)}$, $\begin{bmatrix} \mu^{(1)} \\ \mathbf{w}^{(1)} \end{bmatrix}$, is a multivariate normal distribution with mean:

$$\left(\begin{bmatrix} \mathbf{1} & g(\mathbf{Z}) \end{bmatrix}' \begin{bmatrix} \mathbf{1} & g(\mathbf{Z}) \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{1} & g(\mathbf{Z}) \end{bmatrix}' \mathbf{y} \tag{7}$$

and covariance matrix:

$$\left(\begin{bmatrix} \mathbf{1} & g(\mathbf{Z}) \end{bmatrix}' \begin{bmatrix} \mathbf{1} & g(\mathbf{Z}) \end{bmatrix} \right)^{-1} \sigma_e^2 \tag{8}$$

where $\mathbf{y}$ is a vector of n phenotypes for the observed trait, and $\mathbf{Z}$ is a matrix of size n-by-l_1 for hidden nodes.

Sampling hidden nodes using Hamiltonian Monte Carlo

In the round t of HMC, the acceptance rate, r, can be expressed as:

$$\begin{aligned}
r &= \frac{p(\mathbf{z}_i^*, \boldsymbol{\phi}^* | y_i, ELSE)}{p(\mathbf{z}_i^{[t-1]}, \boldsymbol{\phi}^{[t-1]} | y_i, ELSE)} \\
&= \frac{p(\mathbf{z}_i^* | y_i, ELSE) p(\boldsymbol{\phi}^*)}{p(\mathbf{z}_i^{[t-1]} | y_i, ELSE) p(\boldsymbol{\phi}^{[t-1]})} \\
&= e^{log(p(\mathbf{z}_i^* | y_i, ELSE) p(\boldsymbol{\phi}^*)) - log(p(\mathbf{z}_i^{[t-1]} | y_i, ELSE) p(\boldsymbol{\phi}_i^{[t-1]}))}
\end{aligned} \tag{9}$$

As showed in equation (3) and (9), the log full conditional posterior distribution of the hidden nodes and their gradients are required in HMC. Following equation (3), the full conditional

posterior distribution of the hidden nodes for individual i, *i.e.*, $\mathbf{z}_i$, can be expressed as:

$$
\begin{aligned}
&f(\mathbf{z}_i|y_i, ELSE)\\
&\propto f(\mathbf{z}_i|\boldsymbol{\mu}^{(0)}, \mathbf{w}_1^{(0)}, \ldots, \mathbf{w}_{l_0}^{(0)}, x_{i,1}, \ldots, x_{i,l_0}, \boldsymbol{\Sigma})\\
&\quad f(y_i|\mu^{(1)}, w_1^{(1)}, \ldots, w_{l_1}^{(1)}, \mathbf{z}_i, \sigma_e^2)\\
&\propto |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{(\mathbf{z}_i - \boldsymbol{\mu}^{(0)} - \sum x_{i,m}\mathbf{w}_m^{(0)})^T \boldsymbol{\Sigma}^{-1}(\mathbf{z}_i - \boldsymbol{\mu}^{(0)} - \sum x_{i,m}\mathbf{w}_m^{(0)})}{2}\right\}\\
&\quad \cdot (\sigma_e^2)^{-\frac{1}{2}} \exp\frac{[y_i - \mu^{(1)} - (\mathbf{w}^{(1)})^T g(\mathbf{z}_i)]^2}{-2\sigma_e^2}.
\end{aligned}
\tag{10}
$$

Thus, the log full conditional posterior distribution of $\mathbf{z}_i$ is:

$$
\begin{aligned}
&logf(\mathbf{z}_i|y_i, ELSE)\\
&\propto -\frac{1}{2}\log(|\boldsymbol{\Sigma}|)\\
&\quad -\frac{(\mathbf{z}_i - \boldsymbol{\mu}^{(0)} - \sum x_{i,m}\mathbf{w}_m^{(0)})^T \boldsymbol{\Sigma}^{-1}(\mathbf{z}_i - \boldsymbol{\mu}^{(0)} - \sum x_{i,m}\mathbf{w}_m^{(0)})}{2}\\
&\quad -\frac{1}{2}\log(\sigma_e^2) - \frac{[y_i - \mu^{(1)} - (\mathbf{w}^{(1)})^T g(\mathbf{z}_i)]^2}{2\sigma_e^2}
\end{aligned}
\tag{11}
$$

Thus, the gradient of the log-full conditional posterior distribution of $\mathbf{z}_i$ is:

$$
\begin{aligned}
&\frac{dlogf(\mathbf{z}_i|y_i, ELSE)}{d\mathbf{z}_i}\\
&\propto -\boldsymbol{\Sigma}^{-1}(\mathbf{z}_i - \boldsymbol{\mu}^{(0)} - \sum_{m=1}^{l_0} x_{i,m}\mathbf{w}_m^{(0)})\\
&\quad + \frac{y_i - \mu^{(1)} - (\mathbf{w}^{(1)})^T g(\mathbf{z}_i)}{\sigma_e^2}\mathbf{w}^{(1)} \circ g'(\mathbf{z}_i),
\end{aligned}
\tag{12}
$$

where the symbol "∘" denotes the element-wise production.

Sampling residual variance of the observed trait

The full conditional posterior distribution of σ_e^2 is a scaled inverse chi-squared distribution with $n + \nu_e$ degrees of freedom and scale parameter $\mathbf{e}'\mathbf{e} + S_e^2$, where the $\mathbf{e}$ is $\mathbf{y} - \mathbf{1}\mu^{(1)} - g(\mathbf{Z})\mathbf{w}^{(1)}$.

Genomic Prediction using real datasets

Prediction accuracies for each real dataset are showed in Table 2. In each of the five species, the prediction accuracies of NN-Bayes with different numbers of hidden nodes were not significantly different under the *t*-test with a significance level of 0.05. Significant differences were found among different priors for marker effects in the pig data.

Table 2 Prediction accuracy of NN-Bayes and linear models for real datasets

	#hidden nodes[a]	Priors	Pig	Soy	Spruce	Switchgrass	Maize
Linear model		RR-BLUP	0.534 (0.04)	0.657 (0.02)	0.413 (0.04)	0.865 (0.03)	0.755 (0.09)
		BayesA	0.560 (0.05)[† b]	0.655 (0.02)	0.410 (0.05)	0.866 (0.03)	0.754 (0.09)
		BayesB	0.562 (0.05)[†]	0.655 (0.02)	0.411 (0.05)	0.866 (0.03)	0.755 (0.09)
		BayesCπ	0.533 (0.04)	0.656 (0.02)	0.412 (0.05)	0.865 (0.03)	0.757 (0.09)
		BayesL	0.527 (0.04)	0.655 (0.02)	0.409 (0.04)	0.864 (0.03)	0.755 (0.09)
NN-Bayes	2	RR-BLUP	0.529 (0.04)	0.660 (0.01)	0.418 (0.05)	0.865 (0.03)	0.756 (0.10)
		BayesA	0.562 (0.05)[†]	0.659 (0.02)	0.420 (0.05)	0.866 (0.03)	0.757 (0.10)
		BayesB	0.568 (0.05)[†]	0.658 (0.02)	0.420 (0.05)	0.865 (0.03)	0.757 (0.10)
		BayesCΠ	0.529 (0.04)	0.660 (0.01)	0.419 (0.05)	0.864 (0.03)	0.758 (0.10)
		BayesL	0.523 (0.05)	0.658 (0.02)	0.420 (0.05)	0.866 (0.03)	0.756 (0.10)
	3	RR-BLUP	0.530 (0.04)	0.662 (0.01)	0.420 (0.05)	0.861 (0.03)	0.757 (0.10)
		BayesA	0.562 (0.05)[†]	0.659 (0.02)	0.419 (0.05)	0.865 (0.03)	0.756 (0.10)
		BayesB	0.565 (0.05)[†]	0.659 (0.02)	0.419 (0.05)	0.862 (0.03)	0.756 (0.10)
		BayesCΠ	0.531 (0.04)	0.662 (0.01)	0.419 (0.05)	0.862 (0.03)	0.756 (0.10)
		BayesL	0.524 (0.04)	0.661 (0.02)	0.419 (0.05)	0.863 (0.03)	0.759 (0.10)
	5	RR-BLUP	0.530 (0.04)	0.663 (0.01)	0.419 (0.05)	0.861 (0.03)	0.760 (0.10)
		BayesA	0.565 (0.05)[†]	0.660 (0.02)	0.420 (0.05)	0.860 (0.03)	0.760 (0.10)
		BayesB	0.565 (0.05)[†]	0.660 (0.02)	0.418 (0.05)	0.864 (0.03)	0.754 (0.10)
		BayesCΠ	0.529 (0.04)	0.662 (0.01)	0.419 (0.05)	0.857 (0.03)	0.757 (0.10)
		BayesL	0.523 (0.05)	0.662 (0.02)	0.418 (0.05)	0.861 (0.03)	0.757 (0.10)
	10	RR-BLUP	0.531 (0.04)	0.664 (0.02)[‡ c]	0.419 (0.05)	0.859 (0.03)	0.754 (0.10)
		BayesA	0.563 (0.04)[†]	0.660 (0.02)	0.419 (0.05)	0.860 (0.03)	0.753 (0.10)
		BayesB	0.569 (0.04)[†]	0.660 (0.02)	0.418 (0.05)	0.859 (0.03)	0.755 (0.10)
		BayesCΠ	0.530 (0.05)	0.663 (0.02)[‡]	0.418 (0.05)	0.860 (0.03)	0.754 (0.10)
		BayesL	0.524 (0.05)	0.662 (0.02)	0.418 (0.05)	0.859 (0.03)	0.756 (0.10)

[a] #hidden nodes, number of hidden nodes in NN-Bayes. No significant difference among NN-Bayes with difference number of hidden nodes (*i.e.*, 2, 3, 5, 10). Priors, prior used for marker effects. Prediction accuracy, the average of prediction accuracies from 20 replications with standard deviation in the brackets.
[b] For pig dataset, with significance level = 0.05, the priors used for marker effects in BayesA and BayesB had significantly higher prediction accuracy than other priors, as denoted by † symbol.
[c] For soy dataset, with significance level = 0.1, NN-Bayes composed of 10 hidden nodes with priors for marker effects in BayesCΠ or RR-BLUP had significantly higher prediction accuracy than conventional linear models, as denoted by ‡ symbol.