Neural networks applied to pedigree or genomic-enabled prediction

**Proposition 1**

It must be true that quantitative traits are “complex”, in any sense of the word.

Why?
A "complex" trait involves many metabolic pathways: Roche’s Chart

This is sector G5 of Roche.
Proposition 2

It must be true that epistasis is pervasive

Example: the tricarboxylic acid cycle

For this to work: enzymes are needed
Enzymes in the Krebs cycle

- One gene-one enzyme
- One pathway- many enzymes
- One pathway-many genes

Reactions follow a non-linear dynamics (Michaelis-Menten kinetics)
Proposition 3

A phenotype must be the result of a system involving epistasis and non-linearities of all sorts
CAN ONE WRITE A MECHANISTIC MODEL FOR SOMETHING LIKE THAT?

Proposition 4

• It is unlikely that one could arrive to any reasonable mechanistic model satisfactory to understand, explain, learn and predict outcomes

  GENOMICS (QTL)
  PROTEOMICS (P-QTL)
  METABOLOMICS (BOLO-QTL)
  EXPRESSIONOMICS (E-QTL)
  EPIGENOMICS (M-QTL)
  METAGENOMICS (META-QTL)

Need to navigate in an extraordinarily highly dimensional space to understand “genetic architecture”!!!
Welcome to the world of abstractions! Coping with complexity

First assumption: there is a genetic signal and an environmental signal
Second assumption: the joint effect translates into a phenotype

\[ Y = f(G, E) \]

For some unknown function \( f \)

Choices?

\[ Y = G^E \]
\[ Y = E^G \]
\[ Y = G + E + GE \]
\[ Y = (G + E)^{GE} \]
\[ Y = G + E \]

Is an assumption
Is an even stronger assumption

Further, \( G \) is unknown, so has to be inferred from phenotypes and some input set:

Pedigrees
DNA data
RNA data
Pedigrees, DNA, RNA
THE BIGGEST SHOW ON EARTH:
A prevailing view (Hill et al., 2008; Crow, 2010; Hill, 2010)

• Fisher’s theorem of natural selection
• Interactions are second-order effects; likely tiny and hard to detect
• Detectable pistasis probably arises with genes of large effects, unlikely to be observed in outbred populations
• Epistatic systems generate additive variance and “release” it, so why worry?

THE BIGGEST SHOW ON EARTH: POINT-COUNTERPOINT

• Fisher’s theorem of natural selection (Kempthorne, 1978)
  mean”, again a basic epistemological error. On the matter of the role of variance, to say that additive genetic variance is important “since Fisher’s fundamental theorem of natural selection predicts...” is wide of the mark, and again exemplifies an error commonly made in population genetics. Fisher’s theorem, if it is correct, deals with fitness, whatever that is (and
  
• Interactions are second-order effects; likely tiny and hard to detect
  .....perhaps, but there may be many
  
• Detectable epistasis probably arises with genes of large effects, unlikely to be observed in outbred populations
  .....may be the instruments are not adequate?
  
• Epistatic systems generate additive variance and “release” it, so why worry?
  .....if all we get are straight lines (even though the world is round) how can we learn about ‘genetic architecture’ with such lines, if the world is truly round?
THE BIGGEST SHOW ON EARTH
(The additive genetic model)

Can “Genome” the lion be tamed?

Another show: “Les Idiots Savants”
(much less popular)

- If phenotypic prediction is crucial (medicine, precision mating) can exploitation of interaction have added value?
- Ideally, search for machine that
  -- captures additivity (breeding), interaction (medicine)
  -- has reasonably good predictive ability
  -- general and flexible with respect to input data
  -- does not fail if system is linear and non-interacting
SINGLE MARKER REGRESSION
WITH ORDINARY LEAST-SQUARES
n (#number of observations << p (# markers)

“Full model”

\[ y = X\beta + e \]
\[ = (X_1\beta_1 + X_2\beta_2) + e \]

“marked phenotype”

“OLS” is biased if full model holds and one fits “smaller” model (e.g., single marker Regressions)

\[ y = X_i\beta_i + e \]
\[ E(\beta_1|X_1) = (X'_1X_1)^{-1}E(y) \]
\[ = (X'_1X_1)^{-1}[X_1\beta_1 + X_2\beta_2] \]
\[ = \beta_1 + (X'_1X_1)^{-1}X'_1X_2\beta_2 \]

EXTRAORDINARILY NAÏVE, YET....
GWAS FOR PANCREATIC CANCER…
(Nature Genetics)

SINGLE MARKER REGRESSION: A DISASTER

N=100, 1000 binary markers, 5 first are signal, LD~1/3

RELATIVE MEAN-SQUARED ERROR (ALL MARKERS)

RMSE: all markers

15.36 0.01 0.01 0.03 0.00
A (slightly) less naïve form of approximating $G$ is the whole-genome linear model:

$$G = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_p x_p$$

Where the $x$'s are either pedigree relationships, or marker genotype codes or whatever the latest fad in genomic data is:

- Bayes A
- Bayes B
- Bayes C (with or without n)
- Bayesian Lasso
- NON-BAYESIAN REGULARIZED: Lasso, Elastic Net

LEADS TO (EXTRAORDINARILY) SHRUNKEN ESTIMATES OF EFFECTS, BUT GOOD PREDICTIONS OF "TOTAL SIGNAL"
Some genes do not have introns
Some genes are located within introns of other genes
Arguably, one could do better than with linear Bayesian (regularized) linear models!

\[
\begin{align*}
 f(x) &= f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \ldots \\
\end{align*}
\]

- Linear approximation
- Quadratic approximation

\(f^{(n)}(a)\) - \(n\)th order approximation

A VIEW OF LINEAR MODELS
(as employed in q. genetics)

Mathematically, can be viewed as a “local” approximation of a complex process

[Source: FELDMAN and LEWONTIN (1975)
CHEVALET (1994)]
How good are linear and quadratic approximations? A Taylor series provides a local approximation only…

\[ y = g(x) + e \quad g(x) = \sin(x) + \cos(x) \]

1. Sin and cosine function
2. Linear approximation
3. Quadratic approximation
4. Approximations are good at \( x = 0 \)

“TWO-LOCUS” ADDITIVE MODEL

\[ x_1 + x_2 \]

“TWO-LOCUS” EPISTASIS MODEL

\[ x_1 + x_2 + x_1 x_2 \]

Look at the very different contours

Together

THE ADDITIVE MODEL IS NAÏVE AND INFLEXIBLE
Arguably, one can do better than this

A perhaps more universal learning machine:
Regularized Neural Networks
Why and how neural networks entered as approximators of complex functions…
(a non-mathematical argument)
• Brain superior to von Neumann machines in cognitive tasks

• Microchips: nanoseconds, Brain: milliseconds

• ???

→ Brain recognizes familiar objects from unfamiliar angles
→ Key: not speed but organization of processing


Why?

• Tasks distributed over $10^{12}$ neurons
• Interconnected and activated
• Massively parallel
• Neurons adapt and self-organize
• Interconnectivity: up to $10^3$ synaptic connections
Can we attempt to emulate the brain, mathematically?

**Kolmogorov’s Theorem**

For any continuous function $g(x_1,x_2,\ldots,x_p)$ of $p$ variables there exists continuous functions $h_j$ in $[0, 1]$ a continuous function $g$ in $[0, 1]$ such that

$$g_i(x_{i1},x_{i2},\ldots,x_{ip}) = \sum_{q=3}^{2p+1} \left( \sum_{j=1}^{p} w_j h_j(x_{i1},ix_{i2},\ldots,ix_p) \right)$$

The subscript indicates an evaluation on a given configuration of the input
Comments

• The theorem states that a set of functions exists
• The set includes the possibility of all possible JOINT effects (interactions) among inputs on outputs
• It does not guide on the choice of the functions or on the weights
• With noisy data the idea is to estimate the function from inputs and outputs

KOLMOGOROV’S THEOREM CAN BE REPRESENTED AS AN ARTIFICIAL NEURAL NETWORK
Pedigree, markers, sequences, Nuisance variables

The $h$ functions
(4 “neurons”)

The $f$ functions
(2 “neurons”)

TRANSFORMATIONS (“ACTIVATION”) FUNCTIONS NOT SHOWN

**Binary classification**

$wij = \text{connection strength between input } i \text{ and neuron } j$

$w_{ij} = \text{connection strength between hidden neuron input } i \text{ and output neuron } j$

$wij = \text{connection strength between input } i \text{ and neuron } j$

$wij = \text{connection strength between hidden neuron input } i \text{ and output neuron } j$

**Continuous output: relationship to non-parametric regression**

$y_i = \beta_0 + \sum_{j=1}^{\text{# hidden nodes}} \beta_j \left[ \frac{1}{1 + \exp\left(\frac{x_i}{\eta_j}\right)} \right] + e_i$

If # nodes is known ($k$), the number of parameters is:

$1 + k + k(1 + \# \text{x's}) = 1 + k(\# \text{x's} + 2)$

Can overfit if too many hidden nodes
Types of transformation ("activation") functions

- **Linear**
  \[ y = \varphi(x) = x \]

- **Step**
  \[ y = \varphi(x) = \begin{cases} 1 & x \geq b \\ -1 & x < b \end{cases} \]

- **Piece-wise linear**
  \[ y = \varphi(x) = \begin{cases} 1 & x \geq 1/2 \\ -1/2 & -1/2 \leq x < 1/2 \\ 0 & x \leq -1/2 \end{cases} \]

- **Sigmoid (logistic)**
  \[ y = \varphi(x) = \frac{1}{1 + \exp(-ax)} \]
Hyperbolic tangent

\[
\frac{e^x - e^{-x}}{e^x + e^{-x}}
\]

Illustration of a single-neuron model for \textbf{classification} with logistic activation function

1) Collected input into neuron

\[
\varphi(x_1, x_2, x_3) = \frac{1}{1 + \exp(w_0 + w_1x_1 + w_2x_2 + w_3x_3)}
\]

2) Activated input

3) Classification

\[
\begin{cases}
\varphi(x_1, x_2, x_3) > t \text{ Classify as } "1" \\
\varphi(x_1, x_2, x_3) \leq t \text{ Classify as } "0"
\end{cases}
\]
Illustration of a multi-layer model for regression with logistic activation function before emission to the output layer

Algebraically, the model looks like

\[
y = \beta_0 + \beta_1 \frac{1}{1 + \exp\left(\sum_{j=1}^{4} w_{0j} + \sum_{j=1}^{4} w_{1j} x_1 + \sum_{j=1}^{4} w_{2j} x_2 + \sum_{j=1}^{3} w_{3j} x_3 + \sum_{j=1}^{4} w_{4j} x_4\right)} + \beta_2 \frac{1}{1 + \exp\left(\sum_{j=1}^{4} w_{0j}^{[2]} + \sum_{j=1}^{4} w_{1j}^{[2]} x_1 + \sum_{j=1}^{4} w_{2j}^{[2]} x_2 + \sum_{j=3}^{4} w_{3j}^{[2]} x_3 + \sum_{j=4}^{4} w_{4j}^{[2]} x_4\right)} + \beta_3 \frac{1}{1 + \exp\left(\sum_{j=1}^{4} w_{0j}^{[3]} + \sum_{j=1}^{4} w_{1j}^{[3]} x_1 + \sum_{j=1}^{4} w_{2j}^{[3]} x_2 + \sum_{j=3}^{4} w_{3j}^{[3]} x_3 + \sum_{j=4}^{4} w_{4j}^{[3]} x_4\right)} + \epsilon
\]

4 BETAS + 15 w’s = 19 regressions to estimate
NEURAL NETWORKS ARE UNIVERSAL APPROXIMATORS
(Follows from Kolmogorov’s Theorem)
50 x values sampled from U[-1,1] and then evaluate f(x). Fit a two-layer
NN with 3 hidden nodes and tanh activation functions and linear output

\[
\begin{align*}
\text{Step function} & \\
\text{Output from hidden node} & \\
\end{align*}
\]

Figure 5.3 Illustration of the capability of a multilayer perceptron
to approximate four different functions comprising (a) \(f(x) = e^x\), (b) \(f(x) = \sin(x)\), (c) \(f(x) = |x|\), and (d) \(f(x) = \text{Heaviside step function}\). In each case, \(N = 50\) data points,
shown as blue dots, have been sampled uniformly in \(x\) over the interval
\((-1, 1)\) and the corresponding values of \(f(x)\) evaluated. These data
points are then used to train a two-layer network having 3 hidden units
with ‘tanh’ activation functions and linear output units. The resulting
network functions are shown by the red curves, and the outputs of the
three hidden units are shown by the three dashed curves.

THE INFINITESIMAL MODEL AS A REGRESSION
ON RELATIONSHIPS

\[
\begin{align*}
\mathbf{y} &= \mathbf{u} + \mathbf{e} \\
\mathbf{u} &\sim (0, A\sigma_u^2) \\
\mathbf{y} &= A\mathbf{A}^{-1}\mathbf{u} + \mathbf{e} \\
&= \mathbf{Au}^* + \mathbf{e} \\
y_i &= \sum_{j=1}^{N} a_{ij}u_j^* + e_i
\end{align*}
\]

Use elements of
\(A\) (or \(G\)) as inputs (covariates) in a regression
Model with random effects

Recall
\(A=CC^*\) (Cholesky)
The infinitesimal model as a regression on a pedigree

1) \[ t = Cz \sigma_u + e = Cu^* + e \quad u^* = z \sigma_u \sim (0, I \sigma^2_u) \]
   \[ t_i = g(\sum_{j=1}^n c_{ij} u_j^*) + e_i, \quad \text{Identity activation} \]

2) \[ t = AA^{-1} u + e = Au^{**} + e, \quad u^{**} = A^{-1} u \sim (0, A^{-1} \sigma^2_u) \]
   \[ t_i = g(\sum_{j=1}^n a_{ij} u_{j}^{**}) + e_i, \quad \text{Identity activation} \]

3) \[ t = A^{-1} Au + e = A^{-1} u^{**} + e, \quad u^{***} = Au - (0, A^{-1} \sigma^2_u) \]
   \[ t_i = g(\sum_{j=1}^n a_{ij} u_{j}^{***}) + e_i, \quad \text{Identity activation} \]

The infinitesimal model as a linear neural network

The x's variables are the additive relationships of the animal phenotyped to ALL other individuals in the pedigree
Other than a naïve theory (the infinitesimal additive model) nothing precludes using what might be a better approximation (Kolmogorov)

\[ t_i = g(b + \sum_{k=1}^{S} w_k g_k(b_k + \sum_{j=1}^{n} a_{ij} u^{*k} + e_i), \quad i = 1, 2, \ldots, n \]

"Biases" (intercepts)

“Overall” activation function [linear for quantitative traits]

Neuron-specific activation function

Regression on activated emissions

Elements of pedigree (or genomic) relationships

Bayesian regularization (need to cope with p>>n)

\[ p(D \mid b, w, \sigma^2, M) = \prod_{j=1}^{n} N(t_j \mid b, w, \sigma^2, M) \]

Likelihood

A network Architecture (number of neurons and activation functions)

Prior

\[ p(w \mid \sigma^2_w) = N(0, I\sigma^2_w) \]

(This assumes that all w coefficients are shrunken to the same extent. This is probably not a good assumption, but convenient)

Conditional posterior

\[ P(w \mid D, \sigma^2, \sigma^2_w, M) = \frac{P(D \mid w, \sigma^2, M)P(w \mid \sigma^2_w, M)}{P(D \mid \sigma^2, \sigma^2_w, M)} \]
Marginal density of the data (used to assess variance components)

\[ P(D \mid \sigma^2, \sigma_w^2, M) = \int P(D \mid w, \sigma^2, M)P(w \mid \sigma_w^2, M)dw \]

\[ p(D \mid \sigma^2, \sigma_w^2, M) = \left( \frac{1}{2\pi\sigma^2} \right)^{\frac{n}{2}} \left( \frac{1}{2\pi\sigma_w^2} \right)^{\frac{m}{2}} \times \int \exp \left[ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left( t_i - b - \sum_{k=1}^{S} w_k g_k (b_k + \sum_{j=1}^{n} a_{ij} u_{ij}^{*}) \right)^2 \right] dw \]

\[ F(\alpha, \beta) = \beta \sum_{i=1}^{n} \left( t_i - b - \sum_{k=1}^{S} w_k g_k (b_k + \sum_{j=1}^{n} a_{ij} u_{ij}^{*}) \right)^2 + \alpha w^T w = \beta E_D + \alpha E_w \]

“penalized” sum of squares

Laplacian approximation yields

Remember Smith and Graser (1986); Graser et al. (1987); Tempelman and Gianola (1993)

\[ \log[p(D \mid \alpha, \beta, M)] \approx K + \frac{n}{2} \log(\beta) + \frac{m}{2} \log(\alpha) - \beta E_D + \alpha E_w \|w_{\text{MAP}}^{\text{MAP}}\|_{\text{MAP}}^2 \frac{1}{2} \log \|H\|_{\text{MAP}}^2 \]

\[ \alpha_{\text{new}} = \frac{m}{2W_{\text{MAP}}^{\text{MAP}} + \text{tr}H_{\text{MAP}}^{-1}} \]

\[ \beta_{\text{new}} = \frac{n-m+2\alpha_{\text{MAP}}\text{tr}H_{\text{MAP}}^{-1}}{2\sum_{i=1}^{n} \left( t_i - b - \sum_{k=1}^{S} w_k g_k (b_k + \sum_{j=1}^{n} a_{ij} u_{ij}^{*}) + e_i \right)^2} \]

Effective number of parameters

\[ \gamma = m - 2\alpha_{\text{MAP}}\text{tr}H_{\text{MAP}}^{-1} \]
Data

(297 Jersey cows)

- **Target**: Fat Yield Deviation
  Milk Yield Deviation
  Protein Yield Deviation

- **Inputs**: Elements of Relationship Matrix
  (Pedigree or Genomic, or both)

- **Rationale (again)**

\[
y = u + e \\
u \sim (0, \sigma_u^2) \\
y = AA^{-1}u + e \\
= Au^* + e \\
y_{ij} = \sum_{j=1}^{N} a_{ij}u_{j}^* + e_{ij}
\]

Use elements of \(A\) (or \(G\)) as inputs in NN

35,798 SNPs used to build \(G\) as in Van Raden (2008)

---

**DATA**

Descriptive Statistics

<table>
<thead>
<tr>
<th>Variable</th>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>(CV)</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yield_devMilk</td>
<td>297</td>
<td>1513</td>
<td>1821</td>
<td>(120)</td>
<td>-3669</td>
<td>7544</td>
</tr>
<tr>
<td>Yield_devFat</td>
<td>297</td>
<td>73</td>
<td>103</td>
<td>(142)</td>
<td>-187</td>
<td>1209</td>
</tr>
<tr>
<td>Yield_devProt</td>
<td>297</td>
<td>59</td>
<td>59</td>
<td>(101)</td>
<td>-117</td>
<td>267</td>
</tr>
</tbody>
</table>
Fitting the networks (MATLAB)

- **TRAINING (60%)**, **TUNING (20%)** and **TESTING (20%)** sets
- Non-linear regression with Gaussian prior assigned to the weights and Gaussian likelihood
- Given variances, find mode of weights using non-linear optimization method in **TRAINING** set
- Examine performance in the **TUNING** set
- Predictive performance assessed in **TESTING** set
- NN with 1 Neuron and linear activation function is “animal model” with unknown variances

Run 25 times (to get more stable results) with random partitions
Effective number of parameters
(entire data set)

Sum of squared prediction errors in testing set
Illustration of more results

- Using pedigree additive relationships only
RESULTS (Testing set correlations)

<table>
<thead>
<tr>
<th></th>
<th>Linear</th>
<th>1-neur</th>
<th>2-neur</th>
<th>3-neur</th>
<th>4-neur</th>
<th>5-neur</th>
<th>6-neur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fat_deviation</td>
<td>0.11</td>
<td>0.23</td>
<td>0.22</td>
<td>0.22</td>
<td>0.20</td>
<td>0.23</td>
<td>0.27</td>
</tr>
<tr>
<td>Milk_deviation</td>
<td>0.07</td>
<td>0.10</td>
<td>0.08</td>
<td>0.15</td>
<td>0.09</td>
<td>0.13</td>
<td>0.10</td>
</tr>
<tr>
<td>Prot_Deviation</td>
<td>0.02</td>
<td>0.09</td>
<td>0.08</td>
<td>0.10</td>
<td>0.14</td>
<td>0.16</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Results are average of 25 runs for each architecture

EVIDENCE OF OVERFITTING IN TRAINING TEST
Values of weights (regressions) for the linear and “best” NN

Note the differences in number of weights and in their sizes

REGULARIZATION
Distribution of weights for linear and “best” NN architectures

“Total” influence of inputs in neural network

\[ I = \frac{\sum_{j=1}^{S} ABS(w_{ji})}{\sum_{i=1}^{R} \sum_{j=1}^{S} ABS(w_{ji})} \]
WHEAT DATA SET: 599 lines (480 training-119 testing, 50 random repeats)  
1279 binary markers

<table>
<thead>
<tr>
<th>ANN architectures</th>
<th>Linear</th>
<th>1 neuron</th>
<th>2 neurons</th>
<th>3 neurons</th>
<th>4 neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criterion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effective number of parameters</td>
<td>299±5.5</td>
<td>260±6.1</td>
<td>253±5.9</td>
<td>238±5.5</td>
<td>220±2.8</td>
</tr>
<tr>
<td>Correlations in testing set</td>
<td>0.48±0.03</td>
<td>0.54±0.03</td>
<td>0.56±0.02</td>
<td>0.57±0.02</td>
<td>0.59±0.02</td>
</tr>
<tr>
<td>Mean squared error in testing set</td>
<td>0.99±0.04</td>
<td>0.77±0.03</td>
<td>0.74±0.03</td>
<td>0.71±0.02</td>
<td>0.72±0.02</td>
</tr>
</tbody>
</table>

BENCHMARKS: BAYESIAN LASSO 0.50  4 SVM MODELS 0.50-0.58

ANALYSIS IN PROGRESS BY CROSSA ET AL. (CIMMYT)

Maize corn-flowering  
Data used in Crossa et al. (2010)

<table>
<thead>
<tr>
<th>Trait-environment</th>
<th>M-BL</th>
<th>M-RKHS</th>
<th>M-RBFNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS-ASI</td>
<td>0.5425</td>
<td>0.5926</td>
<td>0.5821</td>
</tr>
<tr>
<td>SS-FLF</td>
<td>0.7417</td>
<td>0.6132</td>
<td>0.7460</td>
</tr>
<tr>
<td>SS-FLM</td>
<td>0.7404</td>
<td>0.6453</td>
<td>0.7678</td>
</tr>
<tr>
<td>WW-ASI</td>
<td>0.5153</td>
<td>0.5580</td>
<td>0.5365</td>
</tr>
<tr>
<td>WW-FLF</td>
<td>0.7268</td>
<td>0.5372</td>
<td>0.7869</td>
</tr>
<tr>
<td>WW-FLM</td>
<td>0.7428</td>
<td>0.5743</td>
<td>0.7981</td>
</tr>
<tr>
<td>SS-GY</td>
<td>0.4743</td>
<td>0.5318</td>
<td>0.5174</td>
</tr>
<tr>
<td>WW-GY</td>
<td>0.5634</td>
<td>0.5459</td>
<td>0.5586</td>
</tr>
</tbody>
</table>
Maize disease -
- GLS --
  high density
  55k

<table>
<thead>
<tr>
<th>Sites</th>
<th>M-BL</th>
<th>M-RKHS</th>
<th>M-RBFNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2188</td>
<td>0.2099</td>
<td>0.2604</td>
</tr>
<tr>
<td>2</td>
<td>0.4174</td>
<td>0.4131</td>
<td>0.4308</td>
</tr>
<tr>
<td>3</td>
<td>0.5899</td>
<td>0.5691</td>
<td>0.5823</td>
</tr>
<tr>
<td>4</td>
<td>0.5215</td>
<td>0.5044</td>
<td>0.5058</td>
</tr>
<tr>
<td>5</td>
<td>0.3419</td>
<td>0.3064</td>
<td>0.3442</td>
</tr>
<tr>
<td>6</td>
<td>0.2842</td>
<td>0.2535</td>
<td>0.2775</td>
</tr>
</tbody>
</table>

Maize under 2 level of drought
-- high density 55k

<table>
<thead>
<tr>
<th>Environment</th>
<th>M-BL</th>
<th>M-RKHS</th>
<th>M-RBFNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>GY-Moderate drought</td>
<td>0.6333</td>
<td>0.5591</td>
<td>0.6531</td>
</tr>
<tr>
<td>GY-Severe drought</td>
<td>0.4104</td>
<td>0.3652</td>
<td>0.3910</td>
</tr>
</tbody>
</table>
## Wheat trait 1

<table>
<thead>
<tr>
<th>Sites</th>
<th>M-BL</th>
<th>M-RKHS</th>
<th>M-RBFNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5969</td>
<td><strong>0.6630</strong></td>
<td>0.6581</td>
</tr>
<tr>
<td>2</td>
<td>0.6861</td>
<td><strong>0.7278</strong></td>
<td>0.7069</td>
</tr>
<tr>
<td>3</td>
<td>0.6224</td>
<td><strong>0.6943</strong></td>
<td>0.6866</td>
</tr>
<tr>
<td>4</td>
<td>0.0673</td>
<td>0.1419</td>
<td><strong>0.1840</strong></td>
</tr>
<tr>
<td>5</td>
<td>0.6481</td>
<td><strong>0.6824</strong></td>
<td>0.6744</td>
</tr>
<tr>
<td>6</td>
<td>0.3798</td>
<td><strong>0.4659</strong></td>
<td>0.4586</td>
</tr>
<tr>
<td>7</td>
<td>0.5984</td>
<td>0.6235</td>
<td><strong>0.6284</strong></td>
</tr>
<tr>
<td>8</td>
<td>0.5493</td>
<td>0.6054</td>
<td><strong>0.6100</strong></td>
</tr>
<tr>
<td>9</td>
<td>0.5374</td>
<td>0.5821</td>
<td><strong>0.5827</strong></td>
</tr>
<tr>
<td>10</td>
<td>0.4775</td>
<td><strong>0.5024</strong></td>
<td>0.4274</td>
</tr>
<tr>
<td>11</td>
<td>0.7721</td>
<td>0.7422</td>
<td><strong>0.8039</strong></td>
</tr>
</tbody>
</table>

## Wheat trait 2

<table>
<thead>
<tr>
<th>Site</th>
<th>M-BL</th>
<th>M-RKHS</th>
<th>M-RBFNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4830</td>
<td><strong>0.5216</strong></td>
<td>0.5149</td>
</tr>
<tr>
<td>2</td>
<td>0.6928</td>
<td>0.6753</td>
<td><strong>0.7085</strong></td>
</tr>
<tr>
<td>3</td>
<td>0.2285</td>
<td><strong>0.3889</strong></td>
<td>0.3827</td>
</tr>
<tr>
<td>4</td>
<td>0.4610</td>
<td>0.5508</td>
<td><strong>0.5557</strong></td>
</tr>
<tr>
<td>5</td>
<td>0.7509</td>
<td>0.7147</td>
<td><strong>0.7880</strong></td>
</tr>
<tr>
<td>6</td>
<td>0.8101</td>
<td>0.8031</td>
<td><strong>0.8399</strong></td>
</tr>
<tr>
<td>7</td>
<td>0.4695</td>
<td><strong>0.5374</strong></td>
<td>0.5285</td>
</tr>
<tr>
<td>8</td>
<td>0.8345</td>
<td>0.8261</td>
<td><strong>0.8657</strong></td>
</tr>
</tbody>
</table>
PUNCH LINE:
over 35 trials, the winner is…

M-BL  |  M-RKHS  |  M-RBFNN
--- | --- | ---
14%  |  34%  |  52%
5  |  12  |  18

Any concerns about the predictive ability of non-parametric methods, relative to those that “help to understand genetic architecture”?

Crossa et al. (2012)
TAG-under review
WHAT ABOUT THE BREEDING VALUE?

1. By network design
2. By math

a) Infinitesimal model

\[ y_i = z_i' u \quad \Rightarrow \quad u_i = z_i' \frac{\partial}{\partial z_i} (z_i' u). \]

b) Markers model

\[ y_i = \sum_{j=1}^{p} x_{ij} \beta_j + e_i = x_i' \beta + e_i. \]

Marked breeding value:

\[ x_i' \left[ \frac{\partial}{\partial x_i} (x_i' \beta) \right] = x_i' \beta. \]

c) Neural network with hyperbolic tangent activation function throughout

\[ t_i = b + c g \left[ \sum_{k=1}^{s} w_i g_s(b_i + \sum_{j=1}^{a} p_j u_j^{(r)}) \right] + e_i. \]

\[ BV_i = p_i' \frac{\partial}{\partial p_i} t_i = c g' \left[ \sum_{k=1}^{s} w_i g_s(b_i + \sum_{j=1}^{a} p_j u_j^{(r)}) \right] p_i \sum_{k=1}^{s} w_i g_s'(b_i + \sum_{j=1}^{a} p_j u_j^{(r)}) u^{(r)}. \]
WHAT ABOUT THE IMPORTANCE OF A GIVEN SNP?


**Equation:**

$$I_{SNP_R} = \frac{\sum_{i=1}^{S}|W_{kj}^{(1,1)}|}{\sum_{j=1}^{S}\sum_{k=1}^{R}|W_{kj}^{(1,1)}|} \times 100,$$

---

Fig. 6. Plots for the index values of 798 SNPs as prediction of BMI. The solid line gives the cutoff point separating SNPs with index values larger than 0.45%.
Table 2. Relative importance of SNPs with $I_{2|X_2Y}$ values larger than 0.45% for each of the non-linear networks

<table>
<thead>
<tr>
<th>SNP ID</th>
<th>$I_{SNP}$ (%)</th>
<th>SNP ID</th>
<th>$I_{SNP}$ (%)</th>
<th>SNP ID</th>
<th>$I_{SNP}$ (%)</th>
<th>SNP ID</th>
<th>$I_{SNP}$ (%)</th>
<th>SNP ID</th>
<th>$I_{SNP}$ (%)</th>
<th>SNP ID</th>
<th>$I_{SNP}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>420</td>
<td>0.45</td>
<td>7985</td>
<td>0.46</td>
<td>420</td>
<td>0.45</td>
<td>1513</td>
<td>0.45</td>
<td>5010</td>
<td>0.46</td>
<td>4319</td>
<td>0.46</td>
</tr>
<tr>
<td>7985</td>
<td>0.47</td>
<td>5012</td>
<td>0.48</td>
<td>7985</td>
<td>0.45</td>
<td>4319</td>
<td>0.46</td>
<td>8990</td>
<td>0.48</td>
<td>7985</td>
<td>0.45</td>
</tr>
<tr>
<td>5012</td>
<td>0.47</td>
<td>8590</td>
<td>0.48</td>
<td>7985</td>
<td>0.45</td>
<td>4319</td>
<td>0.46</td>
<td>8990</td>
<td>0.48</td>
<td>7985</td>
<td>0.45</td>
</tr>
<tr>
<td>8590</td>
<td>0.48</td>
<td>4319</td>
<td>0.48</td>
<td>5012</td>
<td>0.48</td>
<td>8590</td>
<td>0.48</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
</tr>
<tr>
<td>348</td>
<td>0.48</td>
<td>4319</td>
<td>0.48</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
</tr>
<tr>
<td>4319</td>
<td>0.48</td>
<td>5010</td>
<td>0.49</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
</tr>
<tr>
<td>5010</td>
<td>0.49</td>
<td>3891</td>
<td>0.49</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
</tr>
<tr>
<td>3891</td>
<td>0.49</td>
<td>472</td>
<td>0.50</td>
<td>3891</td>
<td>0.49</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
<td>348</td>
<td>0.48</td>
</tr>
<tr>
<td>472</td>
<td>0.50</td>
<td>10136</td>
<td>0.52</td>
<td>3891</td>
<td>0.49</td>
<td>348</td>
<td>0.48</td>
<td>10136</td>
<td>0.46</td>
<td>348</td>
<td>0.48</td>
</tr>
<tr>
<td>10136</td>
<td>0.50</td>
<td>10141</td>
<td>0.52</td>
<td>348</td>
<td>0.52</td>
<td>472</td>
<td>0.54</td>
<td>12132</td>
<td>0.59</td>
<td>10141</td>
<td>0.53</td>
</tr>
<tr>
<td>10141</td>
<td>0.52</td>
<td>10141</td>
<td>0.52</td>
<td>472</td>
<td>0.54</td>
<td>12132</td>
<td>0.59</td>
<td>10141</td>
<td>0.53</td>
<td>472</td>
<td>0.54</td>
</tr>
<tr>
<td>472</td>
<td>0.50</td>
<td>10141</td>
<td>0.52</td>
<td>10136</td>
<td>0.52</td>
<td>348</td>
<td>0.52</td>
<td>472</td>
<td>0.54</td>
<td>10136</td>
<td>0.52</td>
</tr>
<tr>
<td>10136</td>
<td>0.50</td>
<td>10141</td>
<td>0.52</td>
<td>348</td>
<td>0.52</td>
<td>472</td>
<td>0.54</td>
<td>12132</td>
<td>0.59</td>
<td>10141</td>
<td>0.53</td>
</tr>
<tr>
<td>10141</td>
<td>0.52</td>
<td>10141</td>
<td>0.52</td>
<td>348</td>
<td>0.52</td>
<td>472</td>
<td>0.54</td>
<td>12132</td>
<td>0.59</td>
<td>10141</td>
<td>0.53</td>
</tr>
<tr>
<td>472</td>
<td>0.50</td>
<td>10141</td>
<td>0.52</td>
<td>348</td>
<td>0.52</td>
<td>472</td>
<td>0.54</td>
<td>12132</td>
<td>0.59</td>
<td>10141</td>
<td>0.53</td>
</tr>
<tr>
<td>10136</td>
<td>0.50</td>
<td>10141</td>
<td>0.52</td>
<td>348</td>
<td>0.52</td>
<td>472</td>
<td>0.54</td>
<td>12132</td>
<td>0.59</td>
<td>10141</td>
<td>0.53</td>
</tr>
</tbody>
</table>

CONCLUSION

- Neural networks: universal approximators
- Need to arrive at suitable architecture (number of layers, number of neurons, choice of activation functions)
- Neural network must be assessed in predictive ability
- Important variables in a network can be detected
- Coefficients do not have obvious interpretation (except in linear networks)
- The infinitesimal model is a naïve network
- The mechanistic value of the additive model is dubious in the face of complexity of biological systems