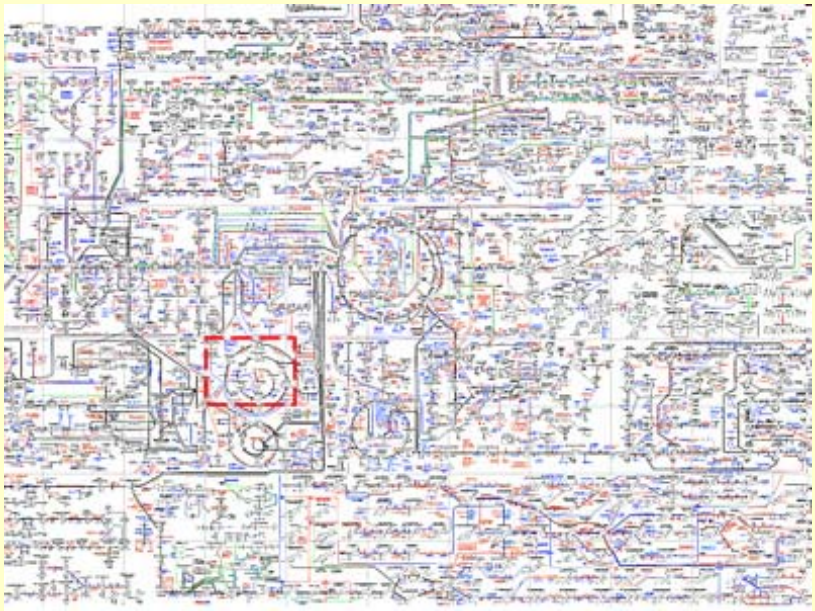
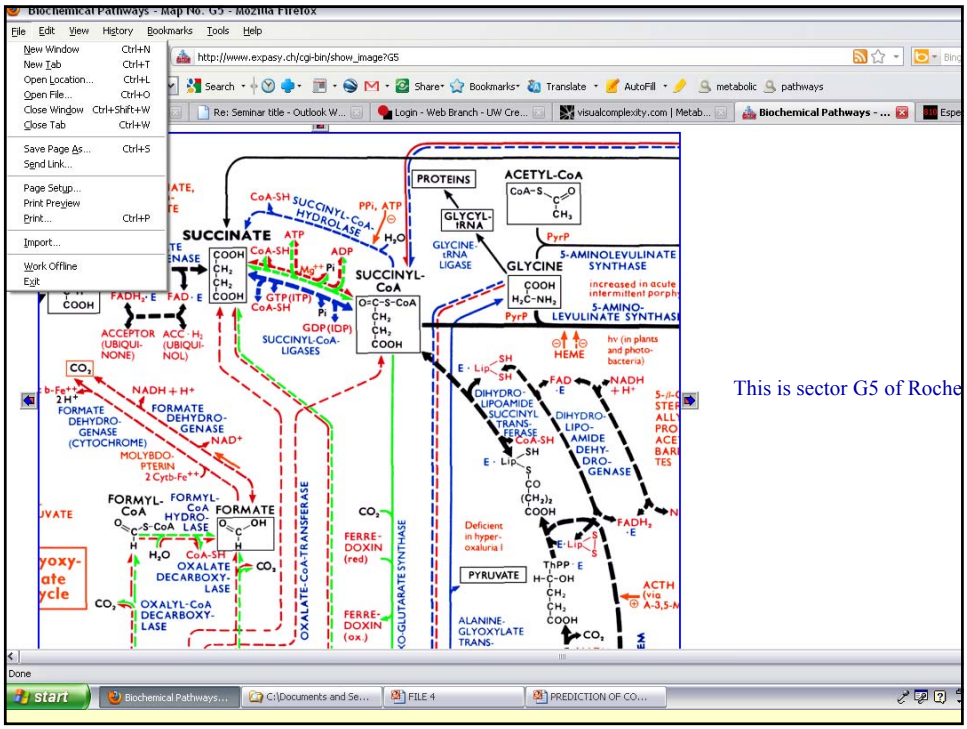
Neural networks applied to pedigree or genomic-enabled prediction

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



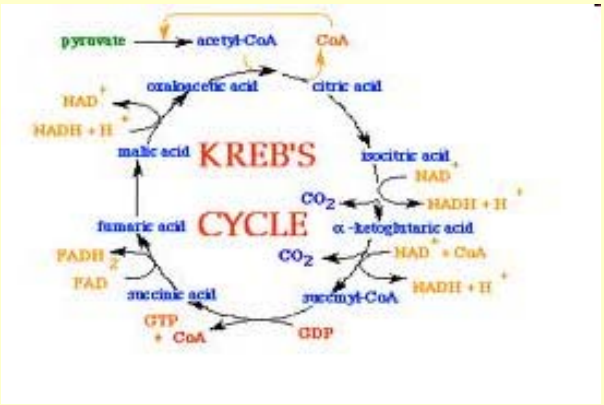


This is sector G5 of Roche

Proposition 2

It must be true that epistasis is pervasive

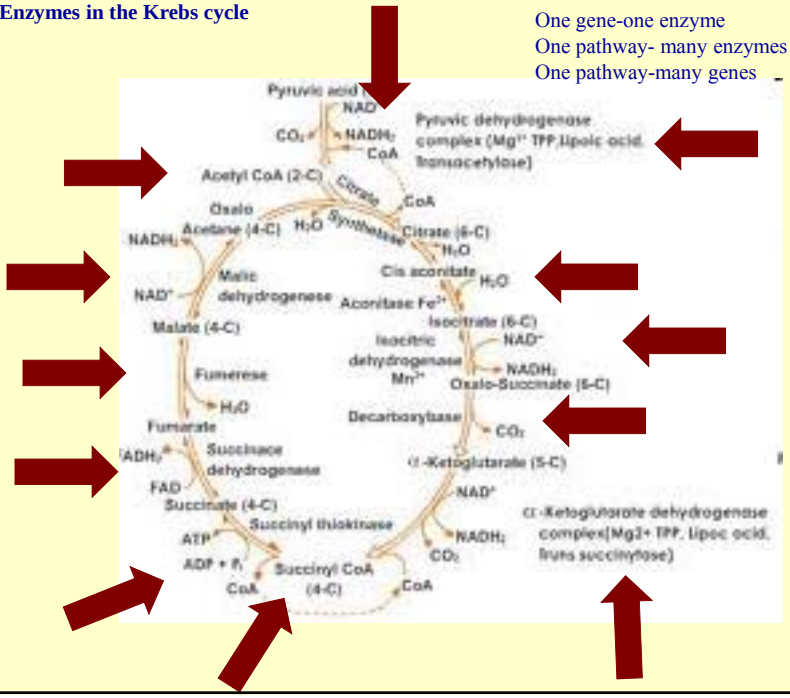
Example: the tricarboxylic acid cycle

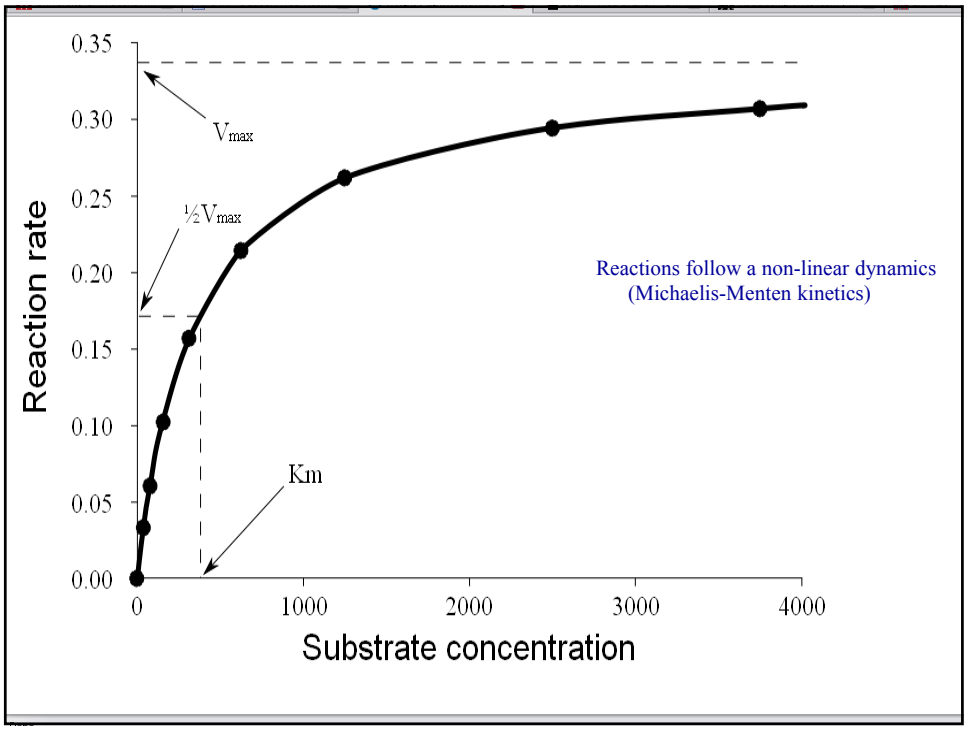


For this to work: enzymes are needed

Enzymes in the Krebs cycle

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



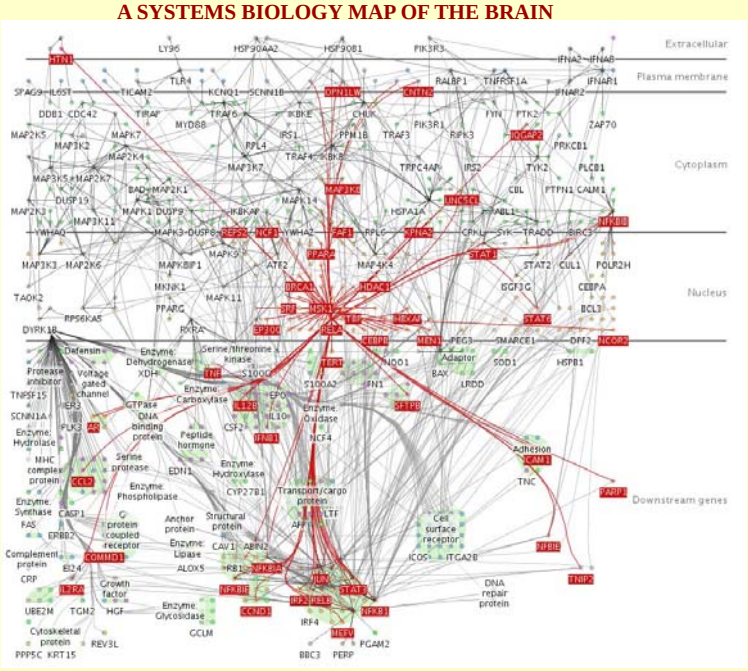


Reactions follow a non-linear dynamics
(Michaelis-Menten kinetics)

Proposition 3

A phenotype must be the result of a system involving epistasis and non-lineariries of all sorts

A SYSTEMS BIOLOGY MAP OF THE BRAIN



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 phenotye **y**

$$Y = f(G,E)$$ For some **UNKNOWN** function *f*

Choices?

- $Y = G^E$?
- $Y = E^G$?
- $Y = G + E + GE$? → Is an assumption
- $Y = (G + E)^{GE}$?
- $Y = G + E$? → Is an even a 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?**

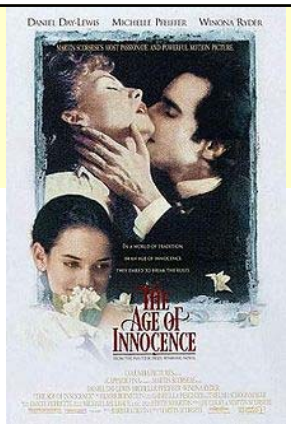
…. if all we get are straight lines (even though the world is round) how can we learn about "genetic architecture" with such lines, if the world is truly round?

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



THE AGE OF INNOCENCE

Unraveling "genetic architecture" with statistical models

SINGLE MARKER REGRESSION
WITH ORDINARY LEAST-SQUARES
n (#number of observations <<p (# markers)

"Full model"

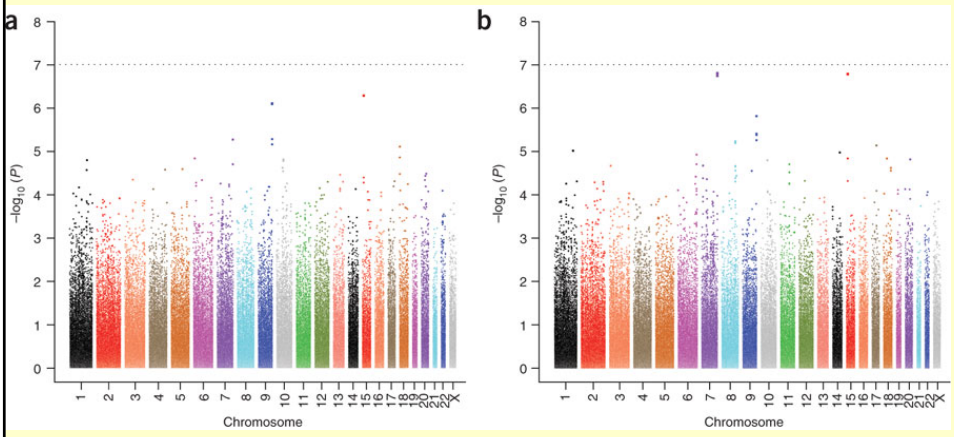
$$y = X\beta + e$$
$$= X_1\beta_1 + X_2\beta_2 + e$$

"marked phenotype"

"OLS" is biased If full model holds and one fits "smaller" model (e.g., single marker Regressions)

$$y = X_1\beta_1 + e$$
$$E(\tilde{\beta}_1|X_1) = (X_1'X_1)^{-1}E(y)$$
$$= (X_1'X_1)^{-1}[X_1\beta_1 + X_2\beta_2]$$
$$= \beta_1 + (X_1'X_1)^{-1}X_1'X_2\beta_2$$

EXTRAORDINARILY NAÏVE, YET….

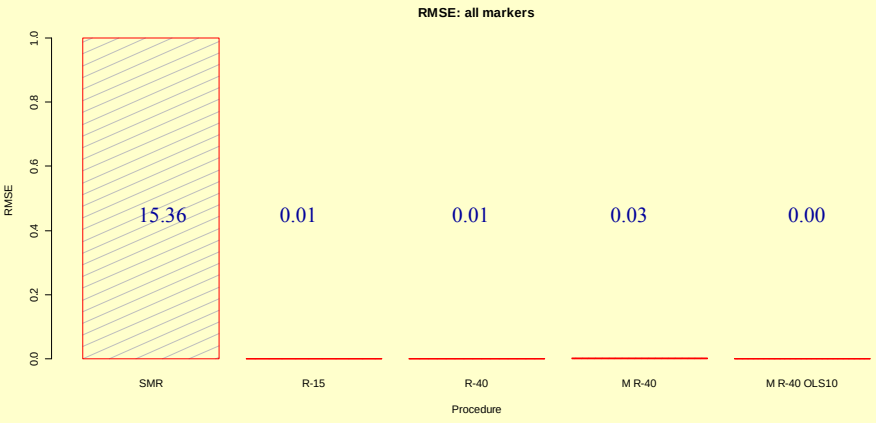


GWAS FOR PANCREATIC CANCER…
(Nature Genetics)

SINGLE MARKER REGRESSION: A DISASTER

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

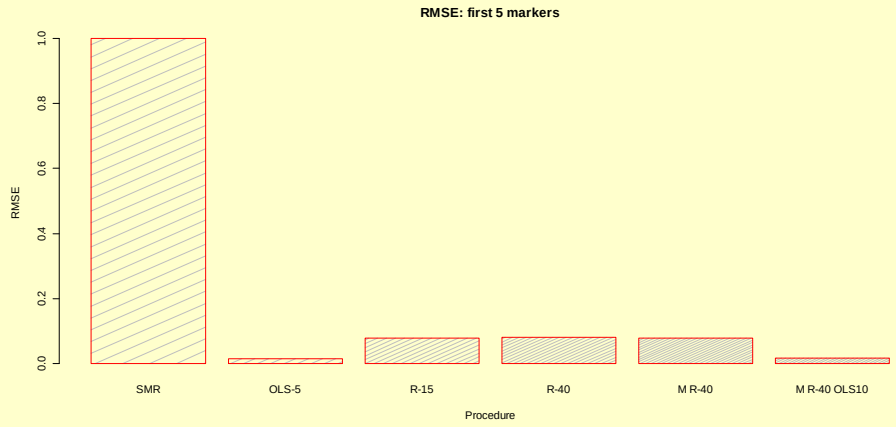
RELATIVE MEAN-SQUARED ERROR (ALL MARKERS)



SINGLE MARKER REGRESSION: A DISASTER

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

RELATIVE MEAN-SQUARED ERROR (FIRST FIVE MARKERS)



A (slightly) less naïve form of approximating G is the whole-genome linear model:

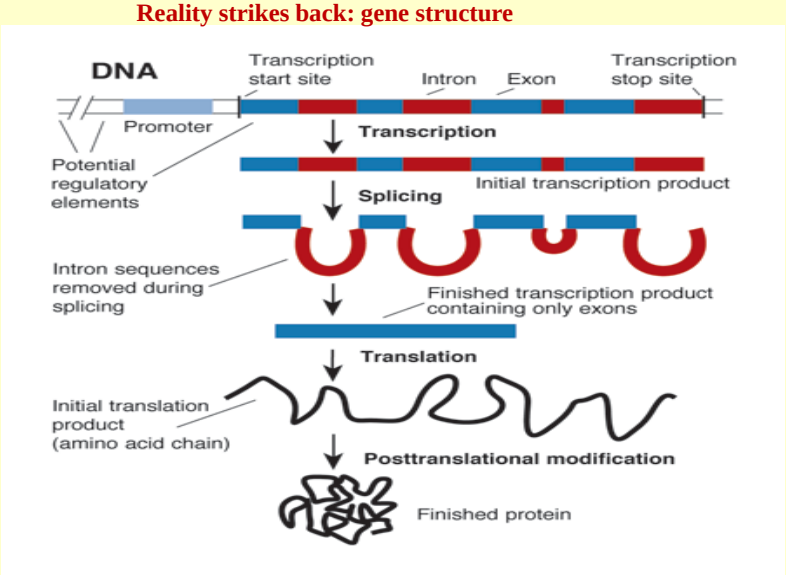
$$G = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_px_p$$

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 π)
Bayesian Lasso
NON-BAYESIAN REGULARIZED: Lasso, Elastic Net

LEADS TO (EXTRAORDINARILY) SHRUNKEN ESTIMATES OF EFFECTS, BUT GOOD PREDICTIONS OF "TOTAL SIGNAL"

Reality strikes back: gene structure



Some genes do not have introns
Some genes are located within introns of other genes

SNPs

Statistical QTL chaser

Gene

Arguably, one could do better than with linear Bayesian (regularized) linear models!

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

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

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \ldots.$$

Linear approximation

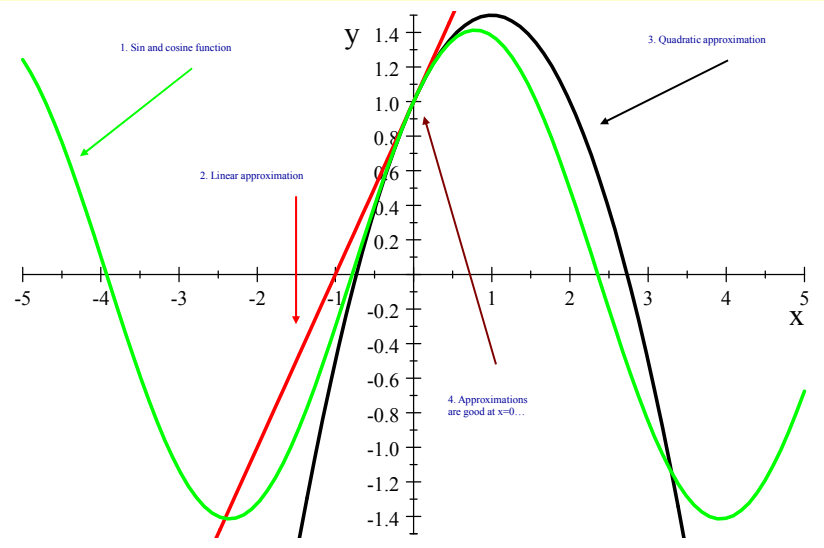
Quadratic approximation

n^{th} order approximation

FELDMAN and LEWONTIN (1975)
CHEVALET (1994)

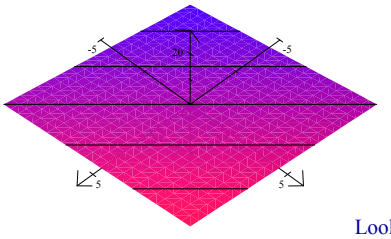
How good are linear and quadratic approximations? A Taylor series provides a local approximation only…

$$y = g(x) + e \qquad g(x) = \sin(x) + \cos(x)$$



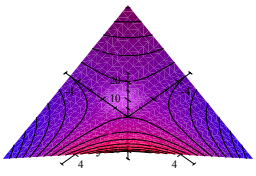
"TWO-LOCUS" ADDITIVE MODEL

$x_1 + x_2$



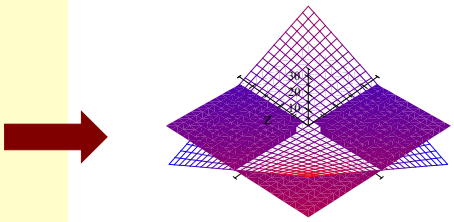
"TWO-LOCUS" EPISTASIS MODEL

$x_1 + x_2 + x_1x_2$



Look at the very different contours

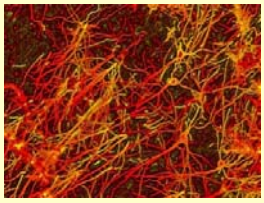
Together

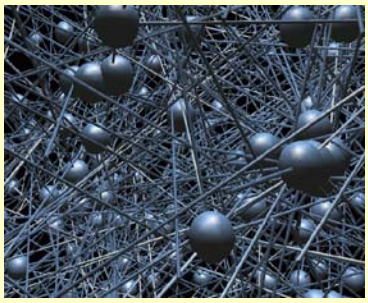


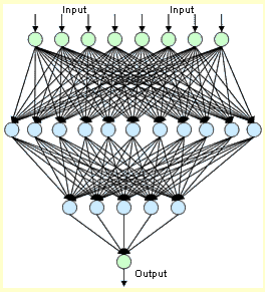
THE ADDITIVE MODEL IS NAÏVE AND INFLEXIBLE

Arguably, one can do better than this

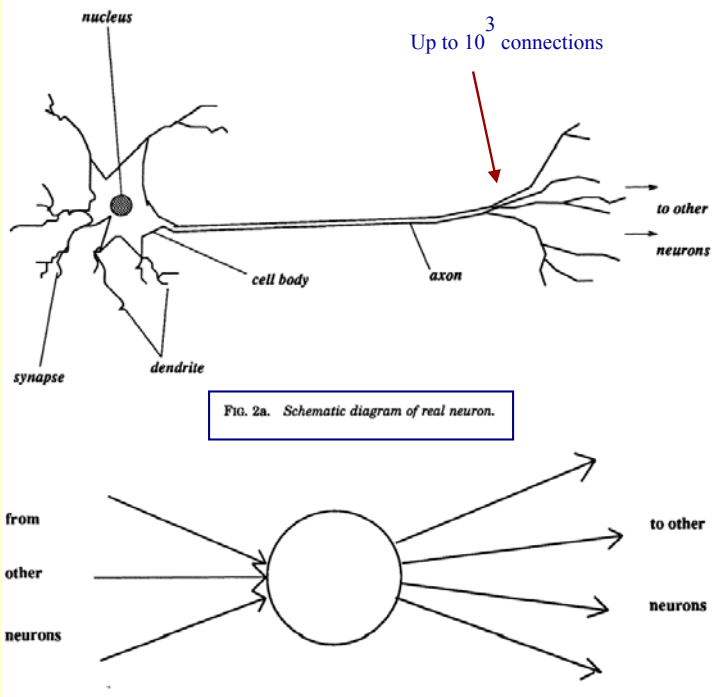
A perhaps more universal learning machine:
Regularized Neural Networks







Why and how neural networks entered as approximators of complex functions…
(a non-mathematical argument)



McCulloch, W. S. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5 115–133.

- 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

$$g_i(x_{i1}, x_{i2}, \ldots, x_{ip}) = \sum_{q=1}^{2p+1} f\left[\sum_{j=1}^{p} w_j h_q(x_{i1}, ix_2, \ldots, ix_p)\right]$$

weights

Linear or nonlinear transformation

Linear or on-linear transformation of inputs

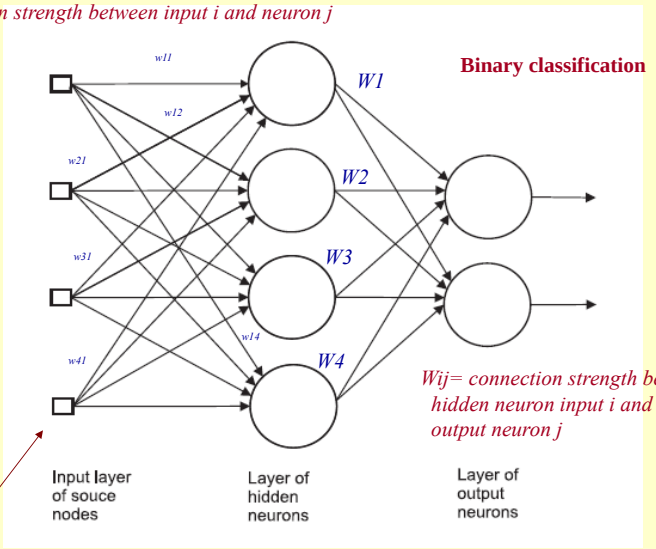
The subscript indicates an evaluation on a given configuration of the input

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 ARTIFICIALNEURAL NETWORK

wij= connection strength between input i and neuron j



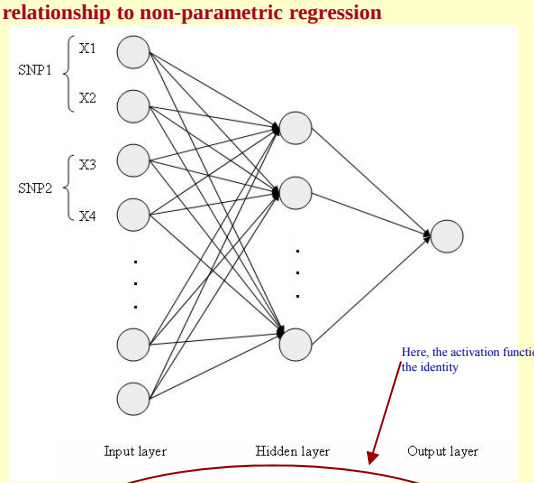
Pedigree, markers, sequences, Nuisance variables

The *h* functions (4 "neurons")

The *f* functions (2 "neurons")

TRANSFORMATIONS ("ACTIVATION") FUNCTIONS NOT SHOWN

Continuous output: relationship to non-parametric regression



$$y_i = \beta_0 + \sum_{j=1}^{\#\text{ hidden nodes}} \beta_j \left[\frac{1}{1+\exp\left(\mathbf{x}_i'\boldsymbol{\gamma}_j\right)} \right] + e_i$$

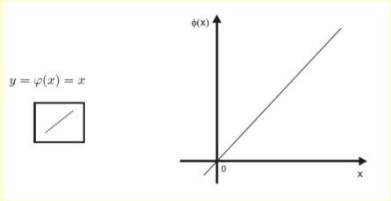
If # nodes is known (k), the number of parameters is:

1+k+k(1+ # x's)= 1+ k (# x's + 2)

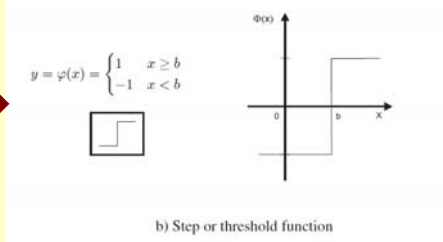
⇒ Can overfit if too many hidden nodes

Types of transformation ("activation") functions

Linear

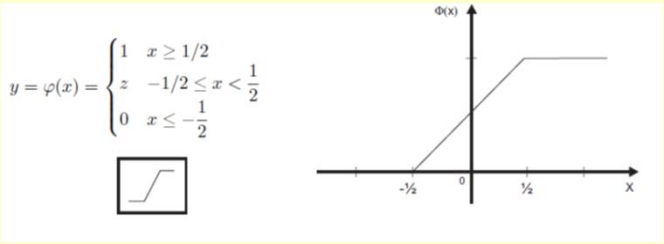


Step

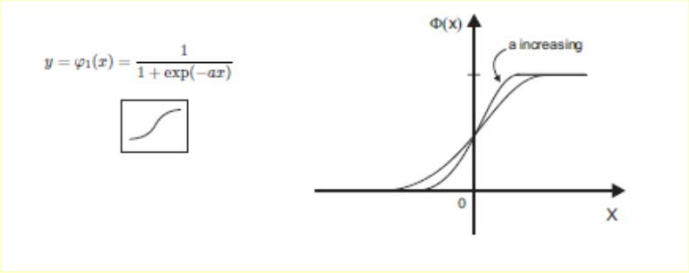


b) Step or threshold function

Piece-wise linear



Sigmoid (logistic)



Hyperbolic tangent

$$(e^{x} - e^{-x})/(e^{x} + e^{-x})$$

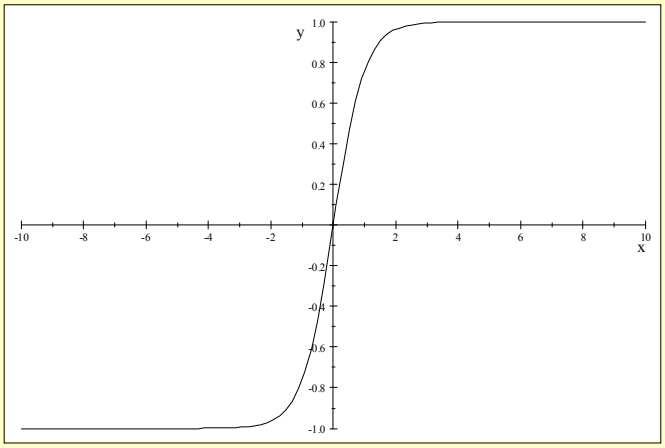
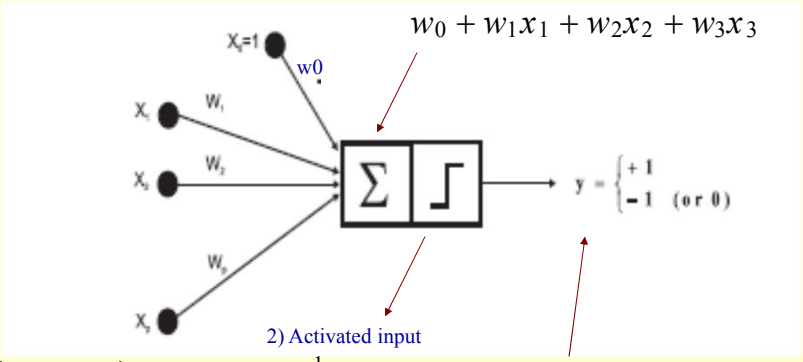


Illustration of a single-neuron model for **classification** with logistic activation function

1) Collected input into neuron

$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

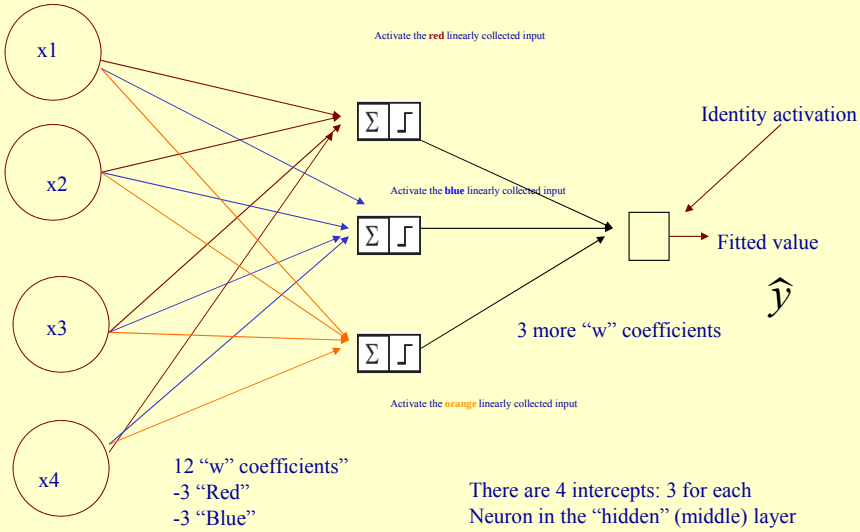


$$\varphi(x_1, x_2, x_3) = \frac{1}{1+\exp(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3)}$$

3) Classification

If $\begin{cases} \varphi(x_1, x_2, x_3) > t \text{ Classify as "1"} \\ \varphi(x_1, x_2, x_3) \leq t \text{ Classify as "0)} \end{cases}$

Illustration of a multi-layer model for **regression** with logistic activation function before emission to the output layer



12 "w" coefficients"
-3 "Red"
-3 "Blue"
-3 "Organge

There are 4 intercepts: 3 for each Neuron in the "hidden" (middle) layer and 1 in the outer layer

Algebraically, the model looks like

$$y = \beta_0 + \beta_1 \frac{1}{1+\exp\left(w_0^{[1]} + w_1^{[1]}x_1 + w_2^{[1]}x_2 + w_3^{[1]}x_3 + w_4^{[1]}x_4\right)} \quad \textbf{RED}$$

$$+ \beta_2 \frac{1}{1+\exp\left(w_0^{[2]} + w_1^{[2]}x_1 + w_2^{[2]}x_2 + w_3^{[2]}x_3 + w_4^{[2]}x_4\right)} \quad \textbf{BLUE}$$

$$+ \beta_3 \frac{1}{1+\exp\left(w_0^{[3]} + w_1^{[3]}x_1 + w_2^{[3]}x_2 + w_3^{[3]}x_3 + w_4^{[3]}x_4\right)} + e$$

ORANGE

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

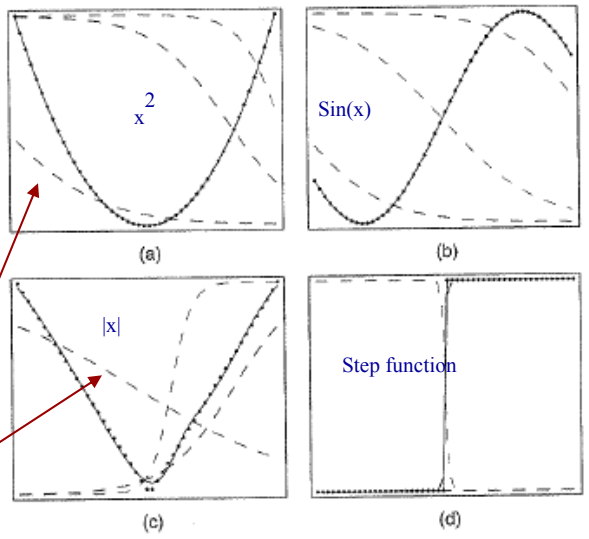


Figure 5.3 Illustration of the capability of a multilayer perceptron to approximate four different functions comprising (a) $f(x) = x^2$, (b) $f(x) = \sin(x)$, (c), $f(x) = |x|$, and (d) $f(x) = H(x)$ where $H(x)$ is the Heaviside step function. In each case, $N = 50$ data points, shown as blue dots, 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.

Output from hidden node

THE INFINITESIMAL MODEL AS A REGRESSION ON RELATIONSHIPS

$$\mathbf{y} = \mathbf{u} + \mathbf{e}$$

$$\mathbf{u} \sim (\mathbf{0}, \mathbf{A}\sigma_a^2)$$

$$\mathbf{y} = \mathbf{A}\mathbf{A}^{-1}\mathbf{u} + \mathbf{e}$$

$$= \mathbf{A}\mathbf{u}^* + e$$

$$y_i = \sum_{j=1}^{N} a_{ij} u_j^* + e_i$$

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) $\mathbf{t} = \mathbf{C}\mathbf{z}\sigma_u + \mathbf{e} = \mathbf{C}\mathbf{u}^* + \mathbf{e}$ $\qquad \mathbf{u}^* = \mathbf{z}\sigma_u \sim (\mathbf{0}, \mathbf{I}\sigma^2{}_u)$

$$t_i = g(\sum_{j=1}^{n} c_{ij} u^*{}_j) + e_i, \qquad \text{Identity activation}$$

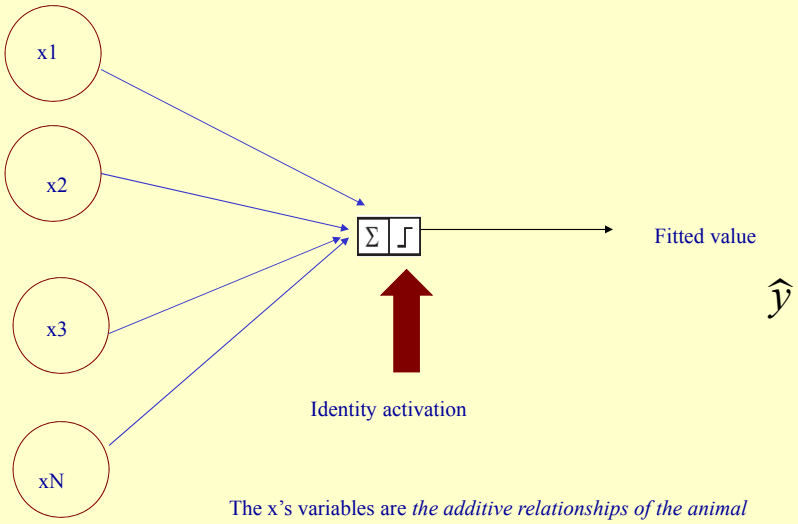
2) $\mathbf{t} = \mathbf{A}\mathbf{A}^{-1}\mathbf{u} + \mathbf{e} = \mathbf{A}\mathbf{u}^{**} + \mathbf{e}, \qquad \mathbf{u}^{**} = \mathbf{A}^{-1}\mathbf{u} \sim (\mathbf{0}, \mathbf{A}^{-1}\sigma^2{}_u)$

$$t_i = g(\sum_{j=1}^{n} a_{ij} u^{**}{}_j) + e_i, \qquad \text{Identity activation}$$

3) $\mathbf{t} = \mathbf{A}^{-1}\mathbf{A}\mathbf{u} + \mathbf{e} = \mathbf{A}^{-1}\mathbf{u}^{**} + \mathbf{e}, \qquad \mathbf{u}^{***} = \mathbf{A}\mathbf{u} \sim (\mathbf{0}, \mathbf{A}^{3}\sigma^2{}_u)$

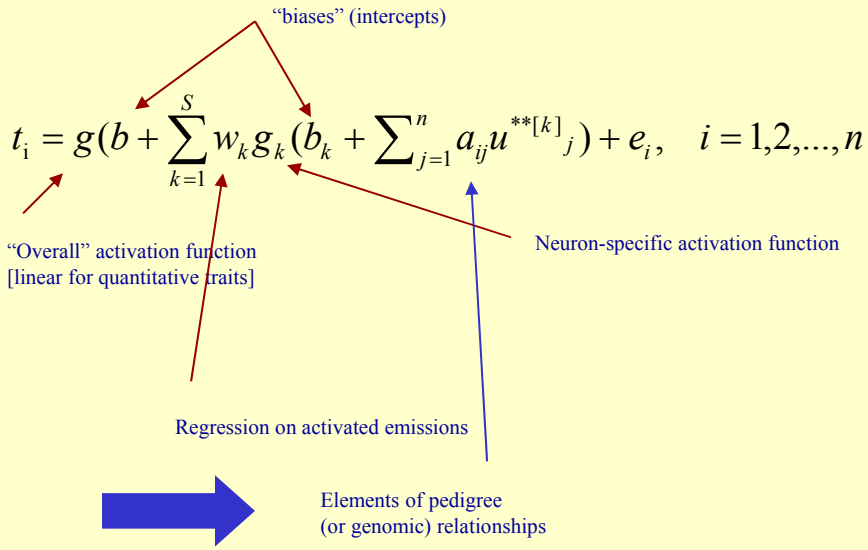
$$t_i = g(\sum_{j=1}^{n} a^{ij} u^{***}{}_j) + e_i, \qquad \text{Identity activation}$$

The infinitesimal model as a linear neural network



The x's variables are *the additive relationships of the animal phenotyed 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)



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

$$p(D \mid b, \mathbf{w}, \sigma^2, M) = \prod_{i=1}^{n} N\left(t_i \mid b, \mathbf{w}, \sigma^2, M\right)$$

Likelihood

A network Architecture (number of neurons and activation functions)

Prior

$$p\left(\mathbf{w} \mid \sigma^2{}_w\right) = N\left(0, \mathbf{I}\sigma^2{}_w\right)$$

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

Conditional posterior

$$P(\mathbf{w} \mid D, \sigma^2, \sigma_w^2, M) = \frac{P(D \mid \mathbf{w}, \sigma^2, M) P(\mathbf{w} \mid \sigma_w^2, M)}{P(D \mid \sigma^2, \sigma_w^2, M)}$$

Marginal density of the data (used to assess variance components)

$$P(D \mid \sigma^2, \sigma_w^2, M) = \int P(D \mid \mathbf{w}, \sigma^2, M) P(\mathbf{w} \mid \sigma_w^2, M) d\mathbf{w}$$

$$p(D \mid \sigma^2, \sigma_w^2, M) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \left(\frac{1}{2\pi\sigma_w{}^2}\right)^{\frac{m}{2}} \times$$

Integral not in closed form in non-linear networks

$$\int \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(t_i - b - \sum_{k=1}^{S} w_k g_k \left(b_k + \sum_{j=1}^{n} a_{ij} u^{**[k]}{}_j \right) \right)^2 - \frac{1}{2\sigma_w{}^2} \mathbf{w}'\mathbf{w} \right] d\mathbf{w}$$

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

"penalized" sum of squares

$1/2\sigma^2$ $1/2\sigma w^2$

Laplacian approximation yields

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

$$\log\left[p(D \mid \alpha, \beta, M)\right] \approx K + \frac{n}{2}\log(\beta) + \frac{m}{2}\log(\alpha) - \left|\beta E_D + \alpha E_w\right|_{w^{map}(\alpha,\beta)} - \frac{1}{2}\log\left\|\mathbf{H}\right\|_{w^{map}(\alpha,\beta)}$$

Hessian of F

$$\alpha_{new} = \frac{m}{2\left(w^{MAP\prime} w^{MAP} + trH_{MAP}^{-1}\right)}$$

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

Effective number of parameters

$$\gamma = m - 2\alpha_{MAP} trH_{MAP}^{-1}$$

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)**

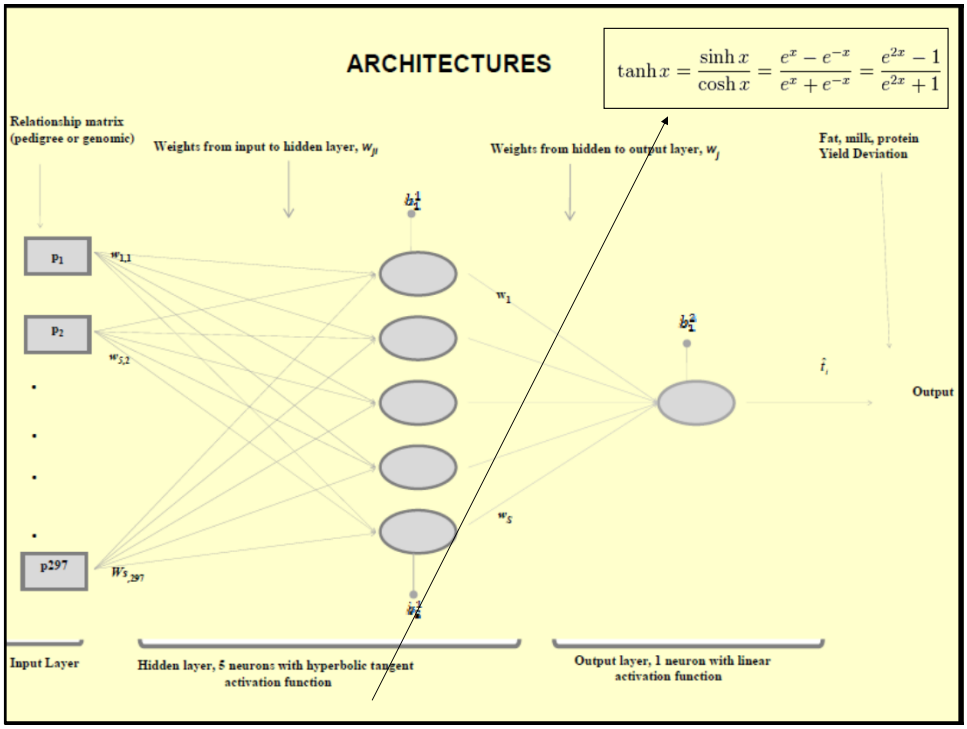
$$\mathbf{y} = \mathbf{u} + \mathbf{e}$$
$$\mathbf{u} \sim (\mathbf{0}, \mathbf{A}\sigma_a^2)$$
$$\mathbf{y} = \mathbf{A}\mathbf{A}^{-1}\mathbf{u} + \mathbf{e}$$
$$= \mathbf{A}\mathbf{u}^* + e$$
$$y_i = \sum_{j=1}^{N} a_{ij} u_j^* + e_i$$

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

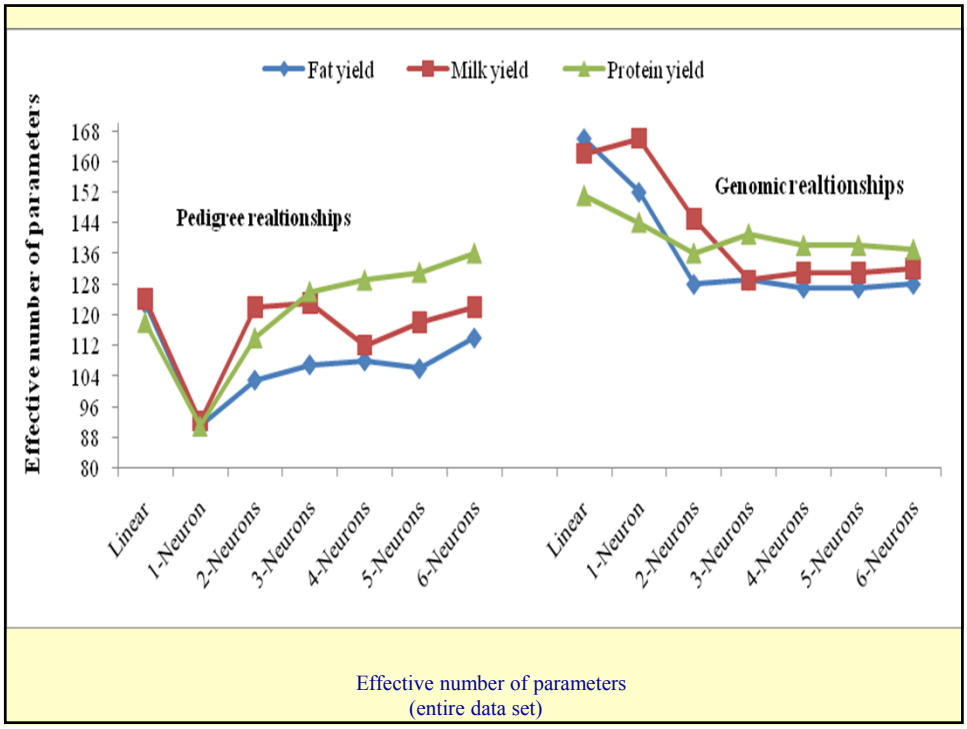
Variable	N	Mean	Std Dev	(CV)	Min	Max
Yield_devMilk	297	1513	1821	(120)	-3669	7544
Yield_devFat	297	73	103	(142)	-187	1209
Yield_devProt	297	59	59.	(101)	-117	267



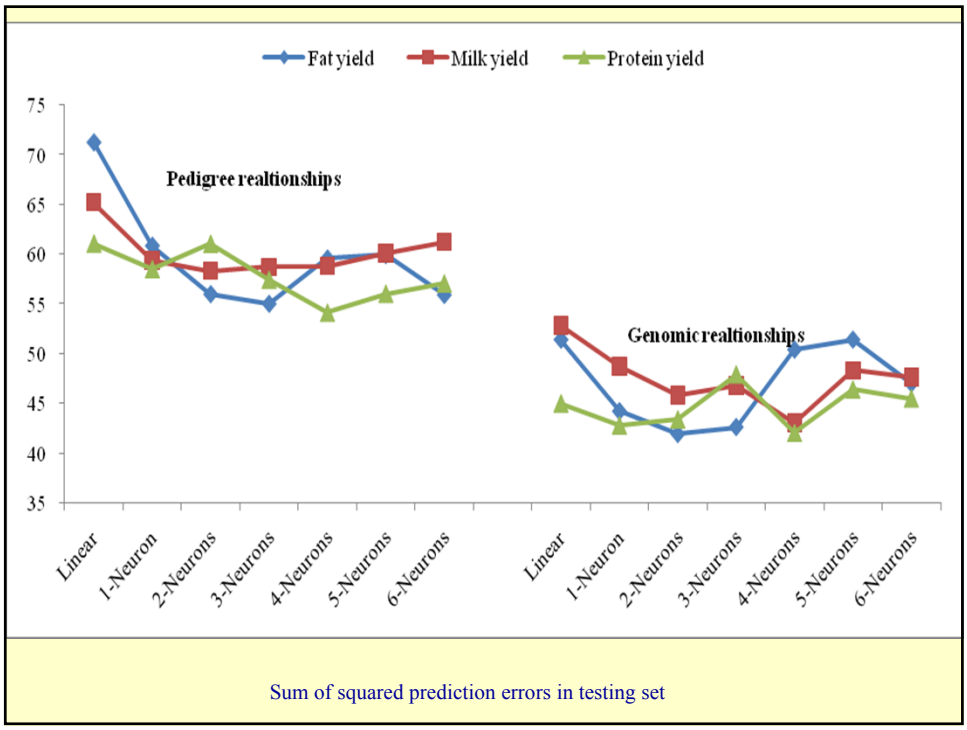
Fitting the networks (MATLAB)

- **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

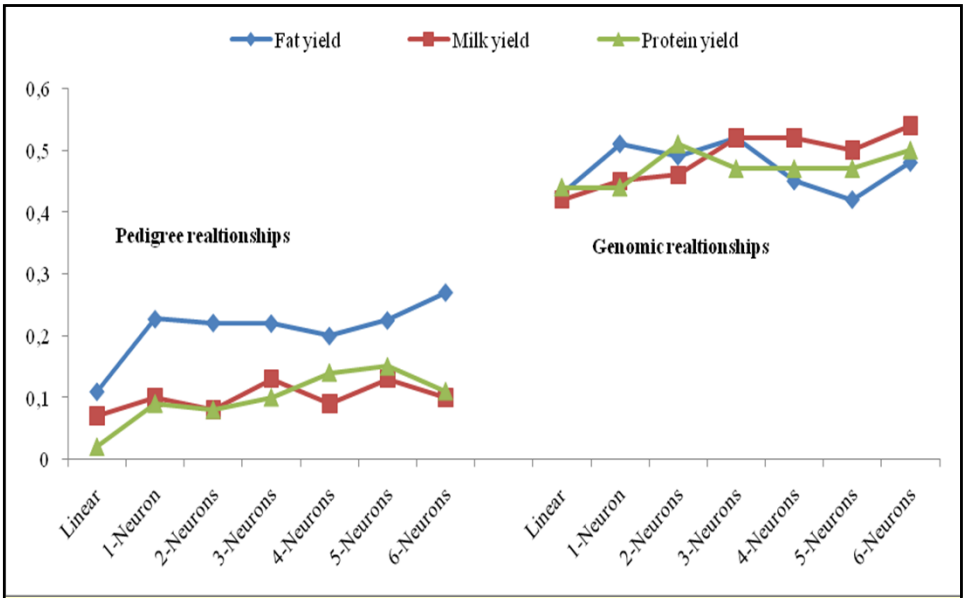
Run 25 times (to get more stable results) with random partitions



Effective number of parameters
(entire data set)



Sum of squared prediction errors in testing set



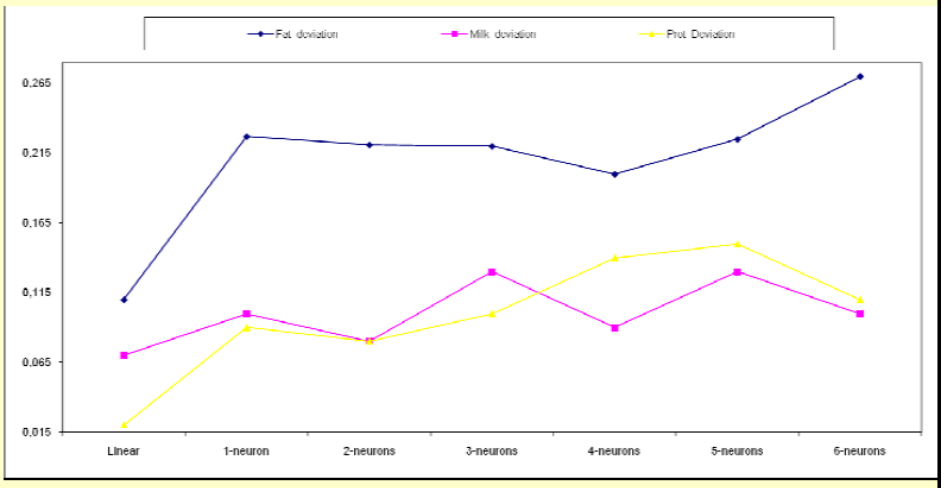
Correlations in testing set

Illustration of more results

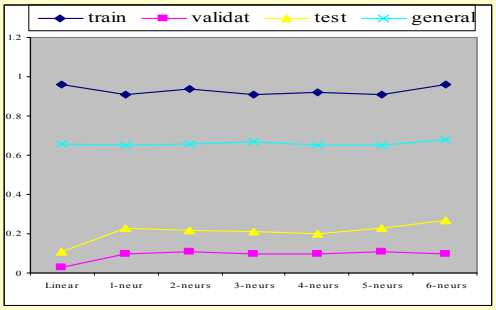
- Using pedigree additive relationships only

RESULTS (Testing set correlations)

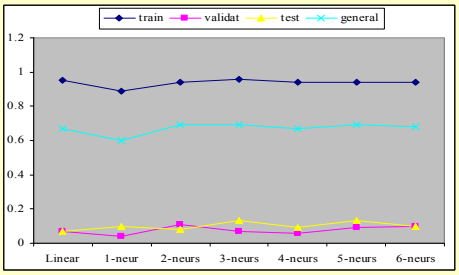
	Linear	1-neuron	2-neurons	3-neurons	4-neurons	5-neurons	6-neurons
Fat_deviation	0,11	0,23	0,22	0,22	0,20	0,23	0,27
Milk_deviation	0,07	0,10	0,08	0,13	0,09	0,13	0,10
Prot_Deviation	0,02	0,09	0,08	0,10	0,14	0,15	0,11

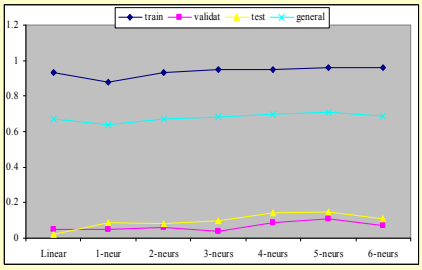


Results are average of 25 runs for each architecture

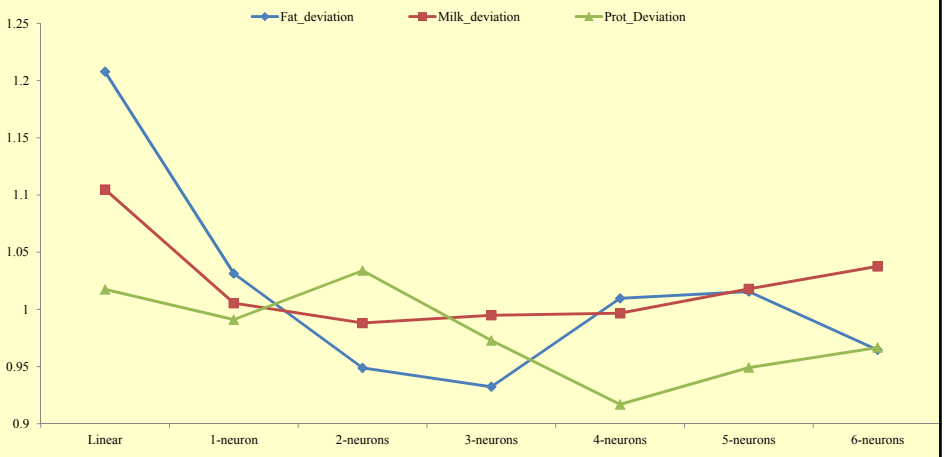


EVIDENCE OF OVERFITTING IN TRAINING TEST

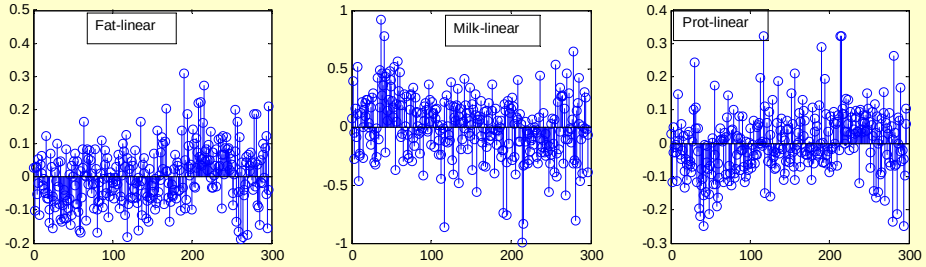




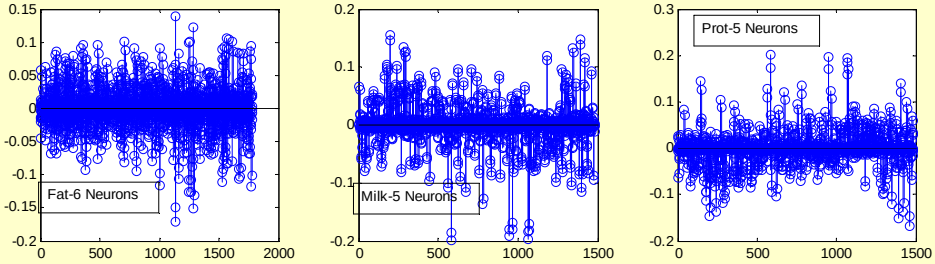
MSE (testing data set)



Values of weights (regressions) for the linear and "best" NN

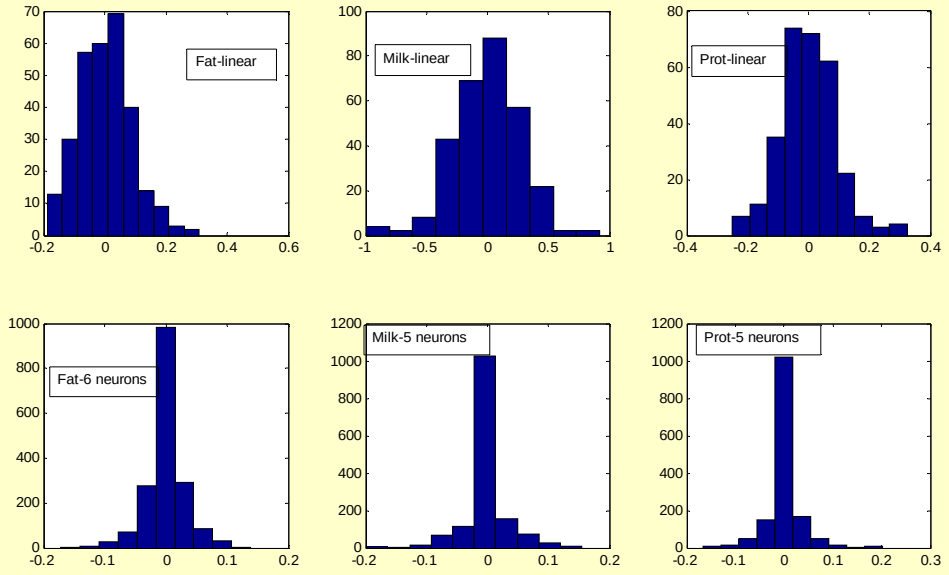


Note the differences in number of weights and in their sizes



REGULARIZATION

Distribution of weights for linear and "best" NN architectures



"Total" influence of inputs in neural network

Value of fat yield deviation	*Anim_id*	*Anim_id*	*Value of fat yield deviation*
212	264	168	252
212	281	194	241
234	261	278	245
241	194	208	265
245	278	214	187
251	255	257	191
252	168	296	256
256	296	211	304
265	208	255	251
304	211	281	212
308	215	215	308
316	190	190	316

NN output

$$I = \frac{\sum_{j=1}^{S} ABS(w_{ji})}{\sum_{i=1}^{R} \sum_{j=1}^{S} ABS(w_{ji})}$$

WHEAT DATA SET: 599 lines (480 training-119 testing, 50 random repeats)
1279 binary markers

ANN architectures	Linear	1 neuron	2 neurons	3 neurons	4 neurons
Criterion					
Effective number of parameters	299±5.5	260±6.1	253±5.9	238±5.5	220±2.8
Correlations in testing set	0.48±0.03	0.54±0.03	056±0.02	0.57±0.02	0.59±0.02
Mean squared error in testing set	0.99±0.04	0.77±0.03	0.74±0.03	0.71±0.02	0.72±0.02

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)

Trait-environment	M-BL	M-RKHS	M-RBFNN
SS-ASI	0.5425	**0.5926**	0.5821
SS-FLF	0.7417	0.6132	**0.7460**
SS-FLM	0.7404	0.6453	**0.7678**
WW-ASI	0.5153	**0.5580**	0.5365
WW-FLF	0.7268	0.5372	**0.7869**
WW-FLM	0.7428	0.5743	**0.7981**
SS-GY	0.4743	**0.5318**	0.5174
WW-GY	**0.5634**	0.5459	0.5586

Maize disease -- GLS -- high density 55k

Sites	M-BL	M-RKHS	M-RBFNN
1	0.2188	0.2099	**0.2604**
2	0.4174	0.4131	**0.4308**
3	**0.5899**	0.5691	0.5823
4	**0.5215**	0.5044	0.5058
5	0.3419	0.3064	**0.3442**
6	**0.2842**	0.2535	0.2775

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

Environment	M-BL	M-RKHS	M-RBFNN
GY-Moderate drought	0.6333	0.5591	**0.6531**
GY-Severe drought	**0.4104**	0.3652	0.3910

Wheat trait 1

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

Wheat trait2

Site	M-BL	M-RKHS	M-RBFNN
1	0.4830	**0.5216**	0.5149
2	0.6928	0.6753	**0.7085**
3	0.2285	**0.3889**	0.3827
4	0.4610	0.5508	**0.5557**
5	0.7509	0.7147	**0.7880**
6	0.8101	0.8031	**0.8399**
7	0.4695	**0.5374**	0.5285
8	0.8345	0.8261	**0.8657**

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

Table 1. Mean correlation of three models, Bayesian LASSO (BL), reproducing kernel Hilbert space (RKHS) regression, and radial basis function neural network (RBFNN), and the number of times one model has a higher correlation than the other (RKHS>BL, RBFNN>BL, and RKHS or RBFNN>BL) for 50 random partitions for each of 23 individual data sets (trait-environment combinations) and across 21 maize data sets.

Trait-environment+	Mean correlation			Number of times a model is better than the other		
	BL	RKHS	RBFNN	RKHS >BL	RBFNN >BL	RKHS >RBFNN
Maize data sets						
FFL-WW	0.814	0.836	0.834	37	32	34
FFL-SS	0.754	0.763	0.757	30	32	22
MFL-WW	0.817	0.841	0.832	37	32	36
MFL-SS	0.776	0.782	0.780	31	36	27
ASI-WW	0.582	0.586	0.594	27	32	23
ASI-SS	0.612	0.621	0.605	34	23	31
GY-SS	0.326	0.330	0.288	28	13	36
GY-WW	0.557	0.548	0.529	16	13	33
GY-HI	0.633	0.663	0.653	37	37	24
GY-LOW	0.410	0.402	0.393	37	31	30
GLS 1	0.220	0.259	0.260	12	20	21
GLS 2	0.419	0.439	0.431	36	17	35
GLS 3	0.590	0.579	0.582	23	25	22
GLS 4	0.522	0.544	0.506	20	24	20
GLS 5	0.346	0.332	0.344	39	38	23
GLS 6	0.284	0.263	0.278	9	25	18
GLS 7	0.477	0.502	0.508	36	16	38
GLS 8	0.596	0.584	0.592	42	29	31
GLS 9	0.522	0.544	0.506	24	21	26
NCBL 1	0.644	0.709	0.691	49	45	40
NCBL 2	0.478	0.491	0.525	34	36	15
Combined 21 maize trait-environment						
	0.542	0.553	0.547	688	627	616

+ FFL: female flowering; MFL: male flowering; ASI: MFL to FFL interval; GY: grain yield; SS: severe drought stress; WW: well-watered environment; HI: optimum environment; LOW: stress environment; GLS: *Cercospora zeae-maydis*; NCLB: *Exserohilum turcicum*.

WHAT ABOUT THE BREEDING VALUE?

1. **By network design**
2. **By math**

a) ***Infinitesimal model***

$$y_i = \mathbf{z'}_i\mathbf{u} \Rightarrow u_i = \mathbf{z'}_i \frac{\partial}{\partial \mathbf{z}_i}(\mathbf{z'}_i\mathbf{u})$$

b) ***Markers model***

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + e_i = \boldsymbol{x}_i'\boldsymbol{\beta} + e_i$$

Marked breeding value=

$$\boldsymbol{x}_i' \frac{\partial}{\partial \boldsymbol{x}_i}(\boldsymbol{x}_i'\boldsymbol{\beta}) = \boldsymbol{x}_i'\boldsymbol{\beta}$$

c) Neural network with hyperbolic tangent activation function throughout

$$t_i = b + cg\left[\sum_{k=1}^{S} w_k g_k\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right)\right] + e_i$$

$$BV_i = \mathbf{p}_i' \frac{\partial}{\partial \mathbf{p}_i} t_i = c\, g'\left[\sum_{k=1}^{S} w_k g_k\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right)\right] \mathbf{p}_i' \sum_{k=1}^{S} w_k g_k'\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right) \mathbf{u}^{**[k]}$$

$$g'\left[\sum_{k=1}^{S} w_k g_k\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right)\right] = 4P(1-P),$$

$$P = \frac{\exp\left[-2\sum_{k=1}^{S} w_k g_k\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right)\right]}{1 + \exp\left[-2\sum_{k=1}^{S} w_k g_k\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right)\right]},$$

$$\mathbf{u}^{**[k]} = \left\{u_j^{**[k]}\right\},$$

and

$$g_k'\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right)\mathbf{u}^{**[k]} = 4P_k(1-P_k),$$

$$P_k = \frac{\exp\left[-2\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right)\right]}{1 + \exp\left[-2\left(b_k + \sum_{j=1}^{n} p_{ij} u_j^{**[k]}\right)\right]}$$

WHAT ABOUT THE IMPORTANCE OF A GIVEN SNP?

Joseph, H., Huang, W. L. & Dickman, M. (2003). Neural network modelling of coastal algal blooms. *Ecology Modelling* **159**, 179–201.

Genet. Res., Camb. (2011), **93**, *pp.* 189–201.
doi:10.1017/S0016672310000662

Prediction of body mass index in mice using dense molecular markers and a regularized neural network

HAYRETTIN OKUT[1,2*], DANIEL GIANOLA[2,3,4], GUILHERME J. M. ROSA[3,4] AND KENT A. WEIGEL[2]

$$I_{\mathrm{SNP}_j} = \frac{\sum_{j=1}^{S} |w_{kj}^{(1,1)}|}{\sum_{j=1}^{S}\sum_{k=1}^{R} |w_{kj}^{(1,1)}|} 100,$$

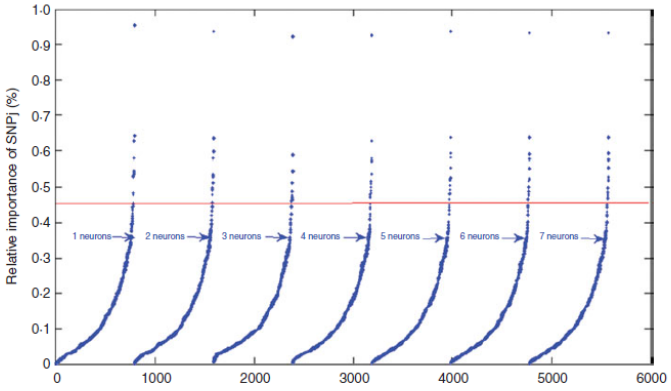


Fig. 6. Plots for the index values of 798 SNPs as prediction of BMI. The solid line gives the cutoff point separating SNPs with index values larger than 0·45%.

Table 2. *Relative importance of SNPs with I_{SNPj} values larger than 0·45% for the each of the non-linear networks*

7 neurons		6 neurons		5 neurons		4 neurons		3 neurons		2 neurons		1 neuron	
SNP ID	I_{SNP} (%)	SNP ID	I_{SNP} (%)	SNP ID	I_{SNP} (%)	SNP ID	I_{SNP} (%)	SNP ID	I_{SNP} (%)	SNP ID	I_{SNP} (%)	SNP ID	I_{SNP} (%)
420	0·45	7985	0·46	420	0·45	1513	0·45	5010	0·46	4319	0·46	1513	0·45
7985	0·47	5012	0·48	7985	0·46	7985	0·45	4319	0·46	8590	0·48	7985	0·45
5012	0·47	8590	0·48	8590	0·48	348	0·48	10 136	0·46	348	0·49	348	0·48
8590	0·48	4319	0·48	5012	0·48	8590	0·48	348	0·47	5012	0·50	8590	0·49
384	0·48	384	0·48	4319	0·48	3891	0·53	10 141	0·47	384	0·50	3891	0·53
4319	0·48	5010	0·49	384	0·48	5012	0·53	472	0·48	5010	0·51	5012	0·53
5010	0·49	3891	0·49	5010	0·49	2487	0·53	3891	0·49	3891	0·51	2487	0·53
3891	0·49	472	0·50	3891	0·49	384	0·54	2487	0·51	472	0·53	384	0·54
472	0·50	10 136	0·52	472	0·50	10 136	0·54	2770	0·54	10 136	0·53	10 136	0·54
10 136	0·52	10 141	0·52	10 136	0·52	5010	0·54	10 961	0·55	2487	0·53	5010	0·54
10 141	0·52	348	0·52	348	0·52	472	0·54	12 132	0·59	10 141	0·53	472	0·54
348	0·53	2487	0·55	10 141	0·53	10 141	0·55	3978	0·92	10 961	0·58	10 141	0·55
2487	0·55	2770	0·58	2487	0·55	10 961	0·58			2770	0·60	10 961	0·58
2770	0·58	10 961	0·59	2770	0·58	2770	0·63			12 132	0·64	2770	0·63
10 961	0·59	12 132	0·64	10 961	0·59	12 132	0·64			3978	0·94	12 132	0·64
12 132	0·64	3978	0·93	12 132	0·64	3978	0·96					3978	0·96
3978	0·93			3978	0·94								

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