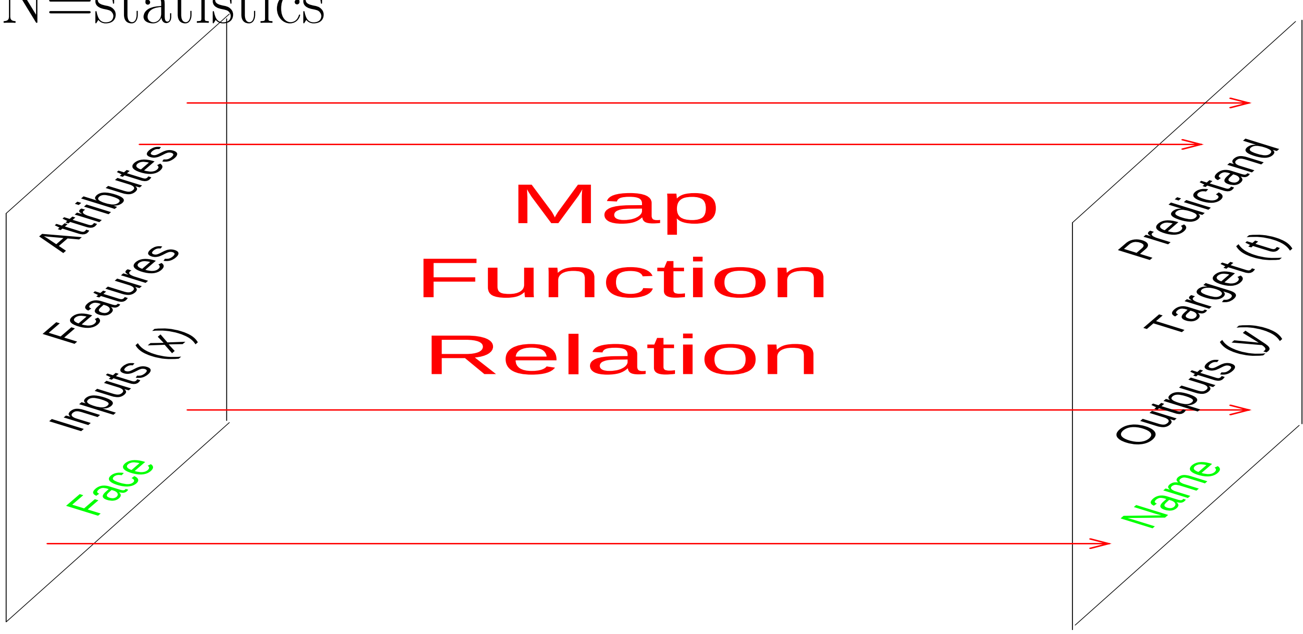
REGRESSION

Caren Marzban
http://www.nhn.ou.edu/˜marzban

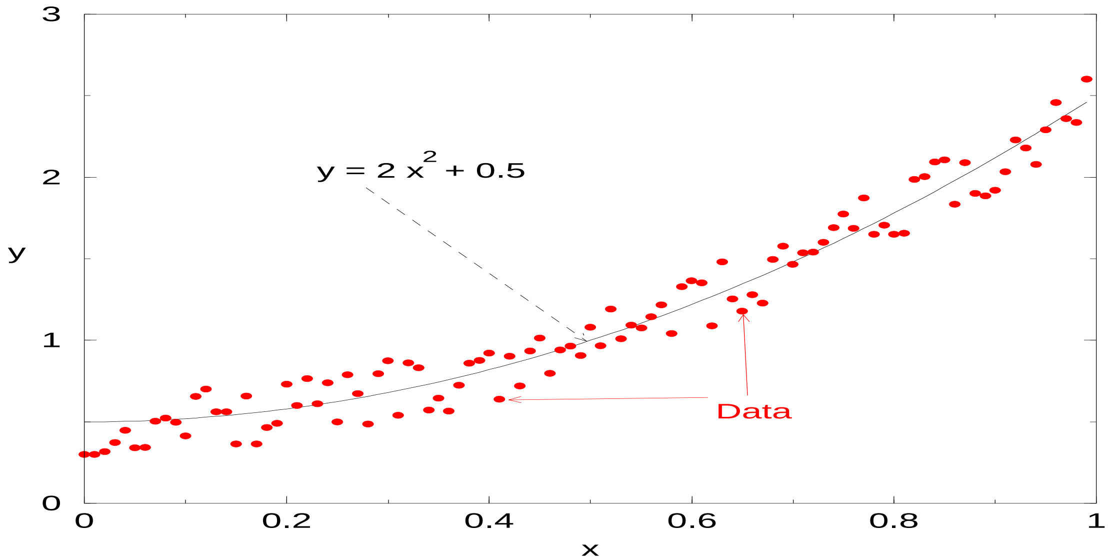
Generalities

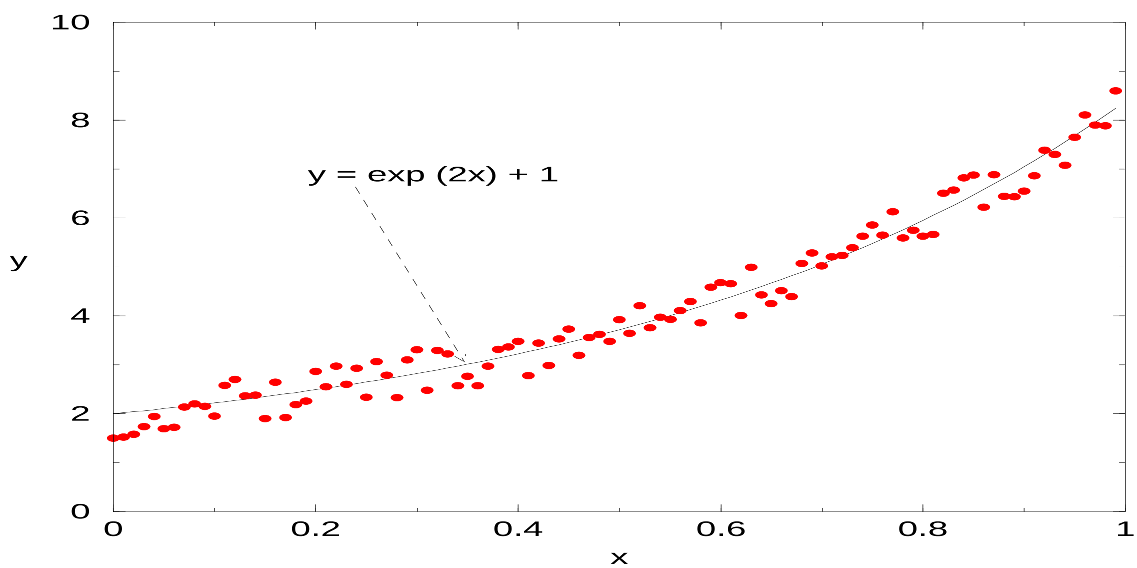
NN=MLP
NN=statistics



Talk 1: Regression.

Continuous target (e.g. temperature ...)





Linear vs. Nonlinear

Note:

0) Regression is about estimating parameters from data.

1) Linear refers to parameters, not x or y:

$$y = \alpha x + \beta$$

2) Some nonlinears are intrinsically linear:

$$y = \exp^{\alpha x} + 1 \quad \rightarrow \quad \log(y - 1) = \alpha x \ .$$

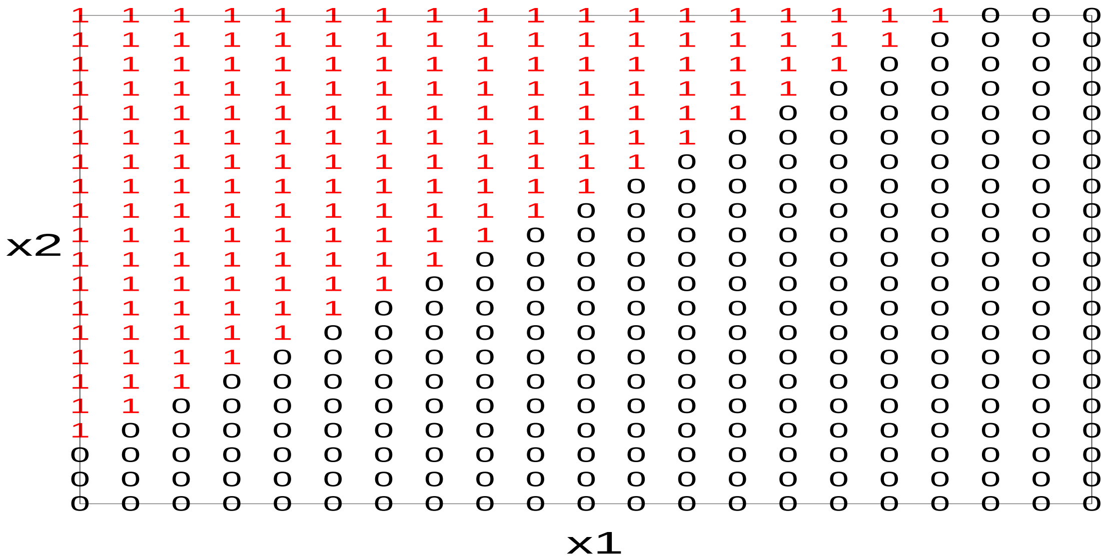
3) Some are not:

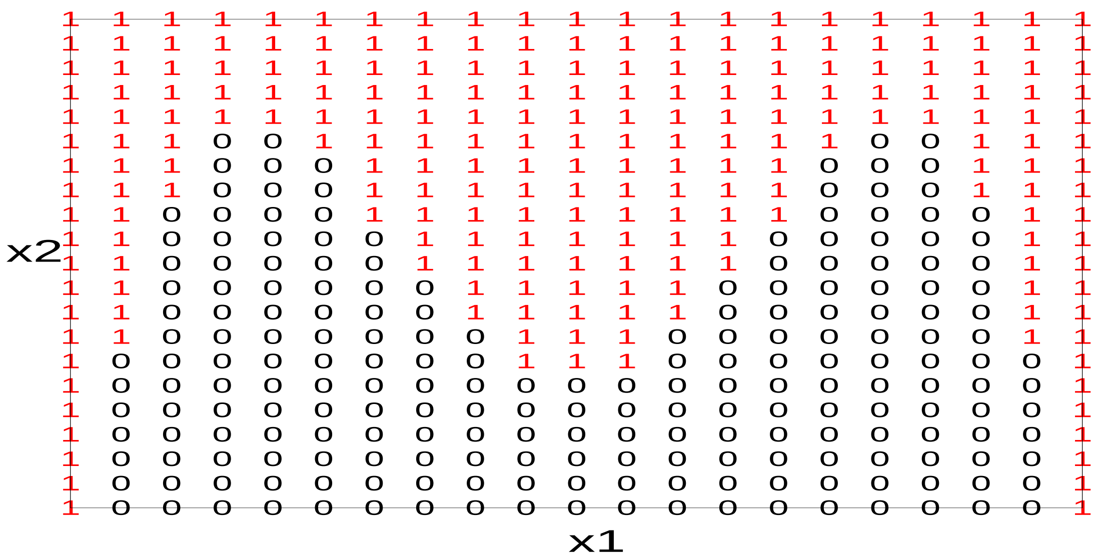
$$y = \alpha \exp^{\alpha x} \ .$$

Talk 2: Classification (Discrimination).

Categorical target (e.g. class=0/1 ...)

Linear and Nonlinear decision boundary.





(Logistic regression = classification!)

Linear Regression

Given data, e.g.,

$$(x_i, t_i), \quad i = 1, 2, 3, ..., N$$

and a model, e.g., $y(x, \omega) = \omega x + \theta$,

$$t_i = \omega x_i + \theta + \epsilon_i$$

how do we estimate the parameters, $\omega,\ \theta$?

Need one more thing, e.g., mean square error (MSE),

$$E(\omega) = \frac{1}{N} \sum_i^N \epsilon_i^2 = \frac{1}{N} \sum_i^N [y(x_i, \omega) - t_i]^2.$$

$$\frac{\partial E}{\partial \theta} = 0 = \frac{\partial E}{\partial \omega}$$

$$\text{(exercise)} \qquad \omega = \frac{< xt > - < x >< t >}{< xx > - < x >< x >}$$

where

$$< xt >= \frac{1}{N} \sum_i^N x_i t_i \qquad \text{etc.}$$

This ω is said to be the Ordinary Least Squares (OLS) estimate.

NN

$$y(x,\omega,H) = g\left(\sum_{i=1}^{H} \omega_i \; f(\sum_{j=1}^{N_{in}} \omega_{ij}x_j - \theta_j) - \omega\right)$$

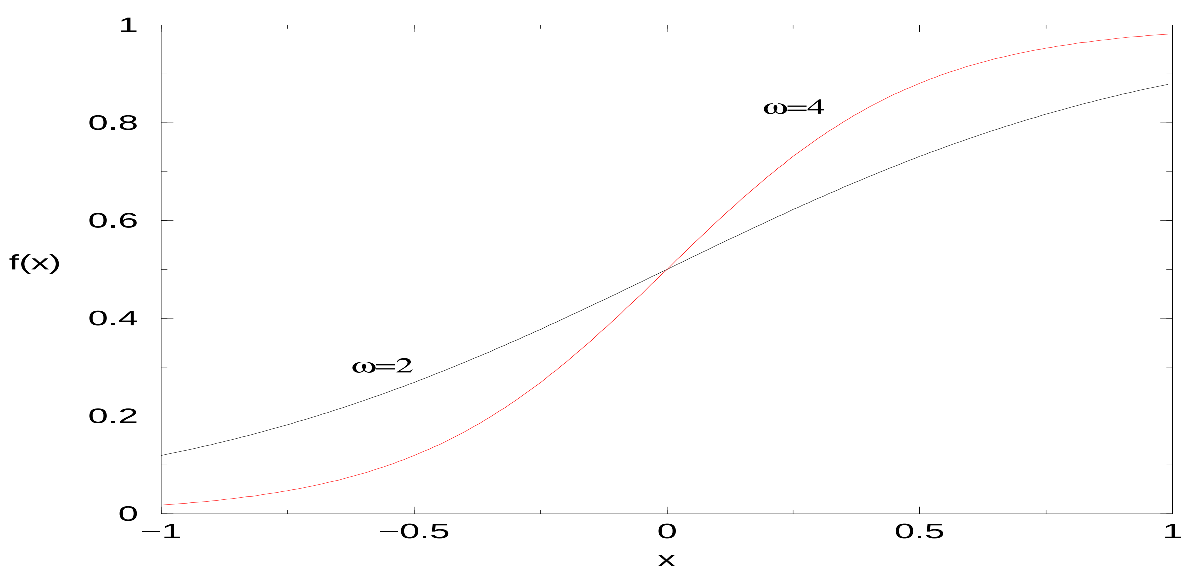
$H = \#$ hdn $\quad f(x) =$ sigmoid $\quad g(x) =$ sigmoid/linear

Recall ...

Nonlinearity $\longrightarrow$ overfitting $\longrightarrow$ poor prediction.

H and $|\omega| \longrightarrow$ nonlinearity.

$H =$ large, $\omega =$ small $\leftrightarrow$ NN $=$ linear $\leftrightarrow$ underfit
$H =$ small, $\omega =$ large $\leftrightarrow$ NN $=$ nonlinear $\leftrightarrow$ overfit



Logistic function $f(x) = \frac{1}{1+e^{-\omega x}}$ for $\omega = 2, 4$.

Question: $\omega = ?$ (training) $H = ?$ (architecture)

$\omega = ?$

$$
\begin{aligned}
P(\omega|D) &\sim P(D|\omega) \times P(\omega) \\
P(\omega|D) &\sim e^{-E_D(\omega)} \times e^{-E_W(\omega)} \\
&\sim e^{-[E_D(\omega)+E_W(\omega)]} \equiv e^{-E(\omega)}
\end{aligned}
$$

Then, max $P(\omega|D) \Longleftrightarrow$ min $E(\omega)$.
Most probable ω, given data, minimizes $E(\omega)$.
Recall MSE.

Specifically, if

$$
\begin{aligned}
P(D|\omega) &\sim e^{-[y(\omega)-t]^2} \text{(gaussian data)} \\
P(\omega) &\sim e^{-\omega^2} \text{(gaussian weights)}
\end{aligned}
$$

then

$$E(\omega) = \Sigma[y(\omega) - t]^2 + \Sigma\omega^2 \sim \text{MSE + Weight-decay}$$

Even in NNs the choice E = MSE assumes normality.
Pick the right E.
Be skeptical of "assumption-free" claims.

Weight-decay caps ω (see H=small,ω=large, above).
Highly recommended.

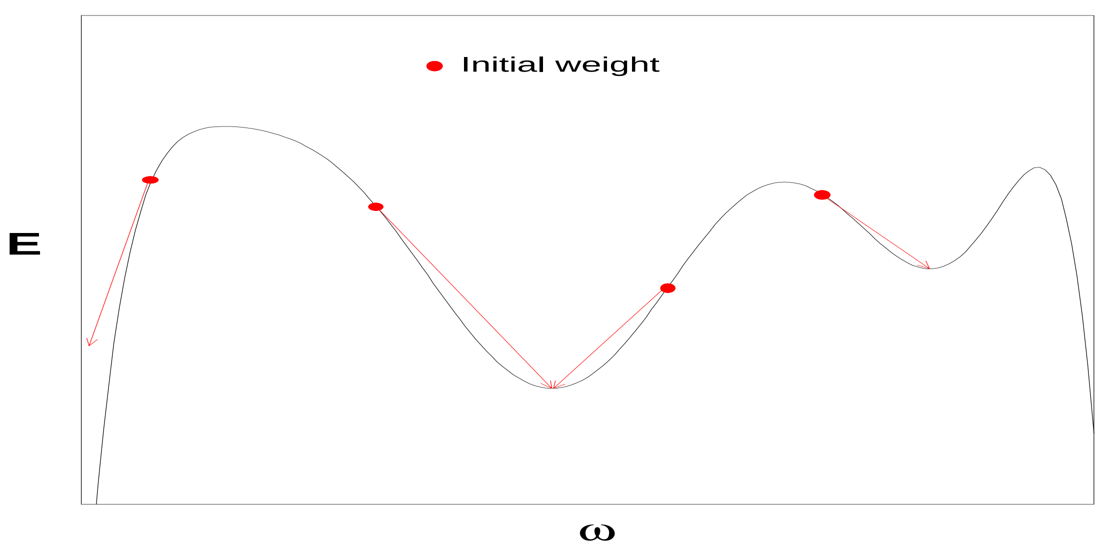
$\omega =?$ (Continued)

$E(\omega) =$ nonlinear in $\omega \rightarrow$ Iterative method.

Dumb:
Go thru all possible ω's,
Calculate E for each ω,
Locate minimum of E,
Select corresponding ω.

Good News:
Gradient decent, BP, conjugate gradient, simulated annealing, genetic algorithm, ...

Bad news: local minima.
Good news: doesn't matter.



Different $\omega_i \rightarrow$ different ω_f, with different/equal $E(\omega_f)$.

$H = ?$

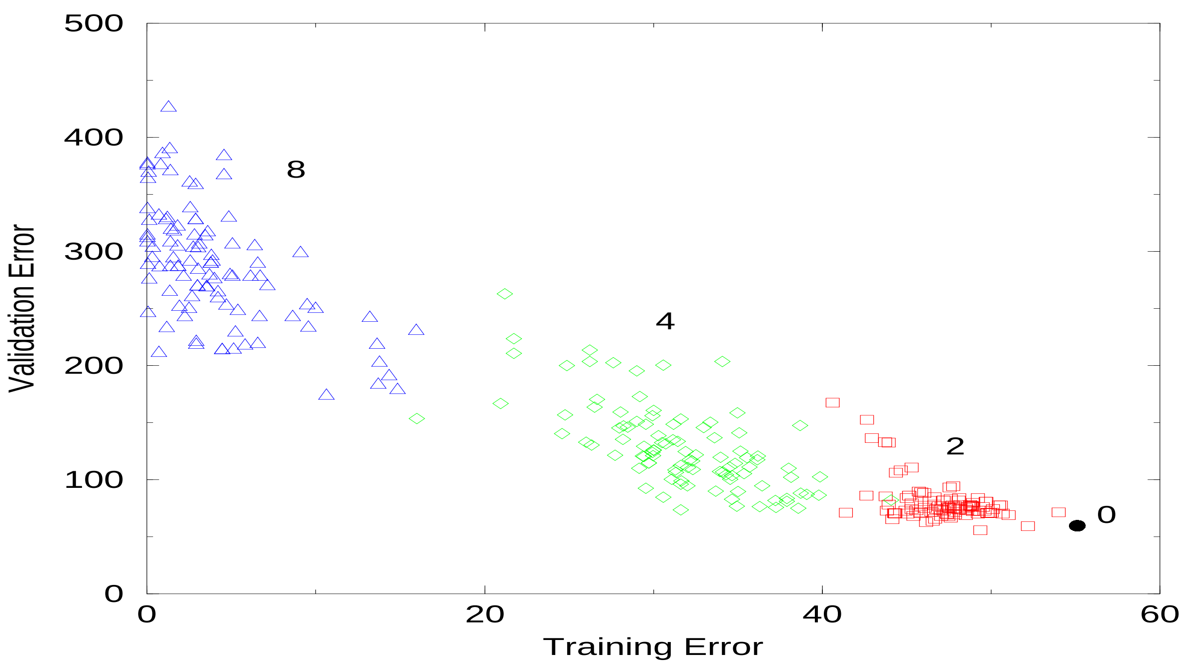
Cross-validation, etc. Bootstrapping.

Data = $\text{trn}_1 \oplus \text{vld}_1$ Trial 1
Data = $\text{trn}_2 \oplus \text{vld}_2$ Trial 2
... ...

Theory suggests the 2/3 rule: $N_{trn} = 2N_{vld}$.

Sometimes need a third set - the test set.

Local minima are intertwined with H.



Training and validation errors at 100 local minima
for NNs with H=0,2,4,8. (tv-diagram)

Different initial ω's (seed) and Bootstrap (trial)
(t,v) = (training error, validation error)

H	Seed	Bootstrap Trial		
		1	2	...
	1	(t,v)	(t,v)	...
2	2	(t,v)	(t,v)	...
	...	(t,v)	(t,v)	...
	1	(t,v)	(t,v)	...
4	2	(t,v)	(t,v)	...
	...	(t,v)	(t,v)	...
	1	(t,v)	(t,v)	...
8	2	(t,v)	(t,v)	...
	...	(t,v)	(t,v)	...
	...	...	...	...
	...	tv-diagram	tv-diagram	...
		$\downarrow$	$\downarrow$	
		H_1	H_2	$\rightarrow H_{\text{optimal}}$

We already know

H = too small $\longrightarrow$ underfit trn set

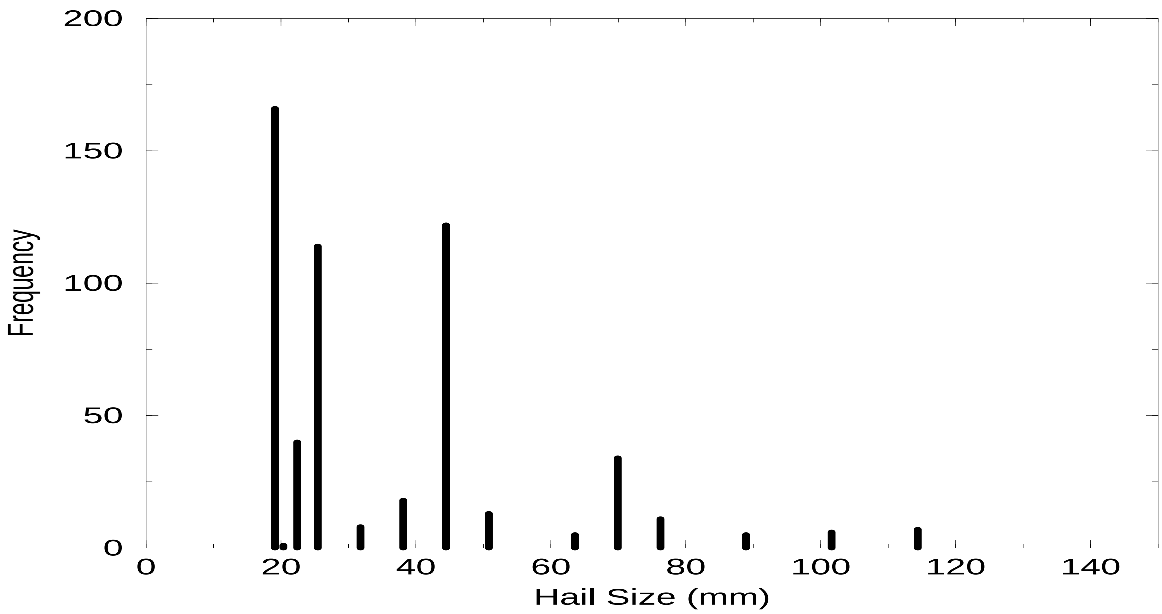
H = too large $\longrightarrow$ overfit trn set

But common mistake

$H = H_1 \longrightarrow$ overfit vld$_1$

One vld error = biased measure of performance

Practical Application - Hail Size



The distribution of hail-size. Note the peaks.

$N_{in} = 9, N_{out} = 1 =$ hail size, $N = 550$

Preprocess:

Examine distribution (histogram) of inputs/output.

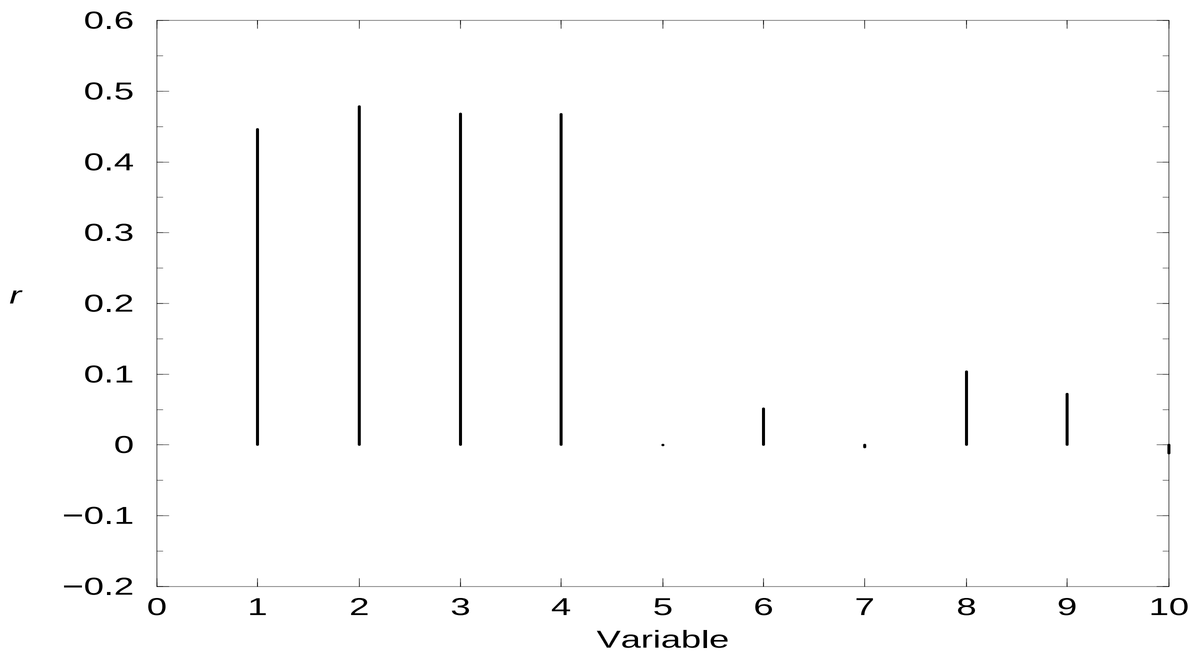
Eliminate outliers from trn set.

Exclude collinear inputs ($15 \longrightarrow 9$).

Transform inputs and target to z-scores:

$$z = (x - \mu)/\sigma.$$

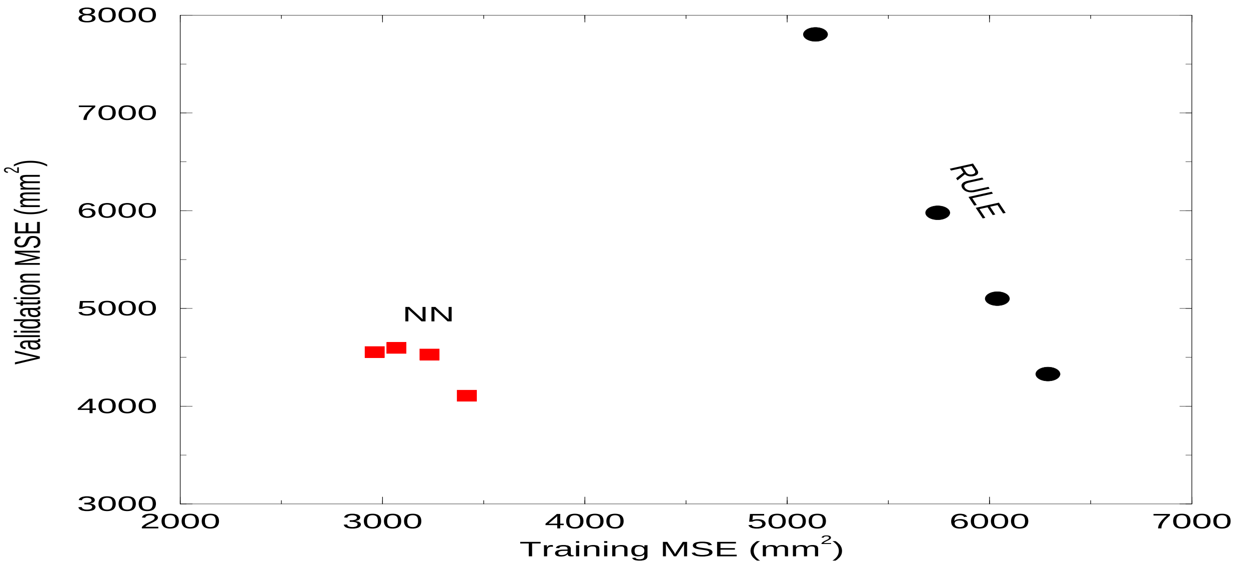
Choose E (=MSE, affected by outliers).
Note: post-check normality of residuals ($t_i - y(x_i)$).

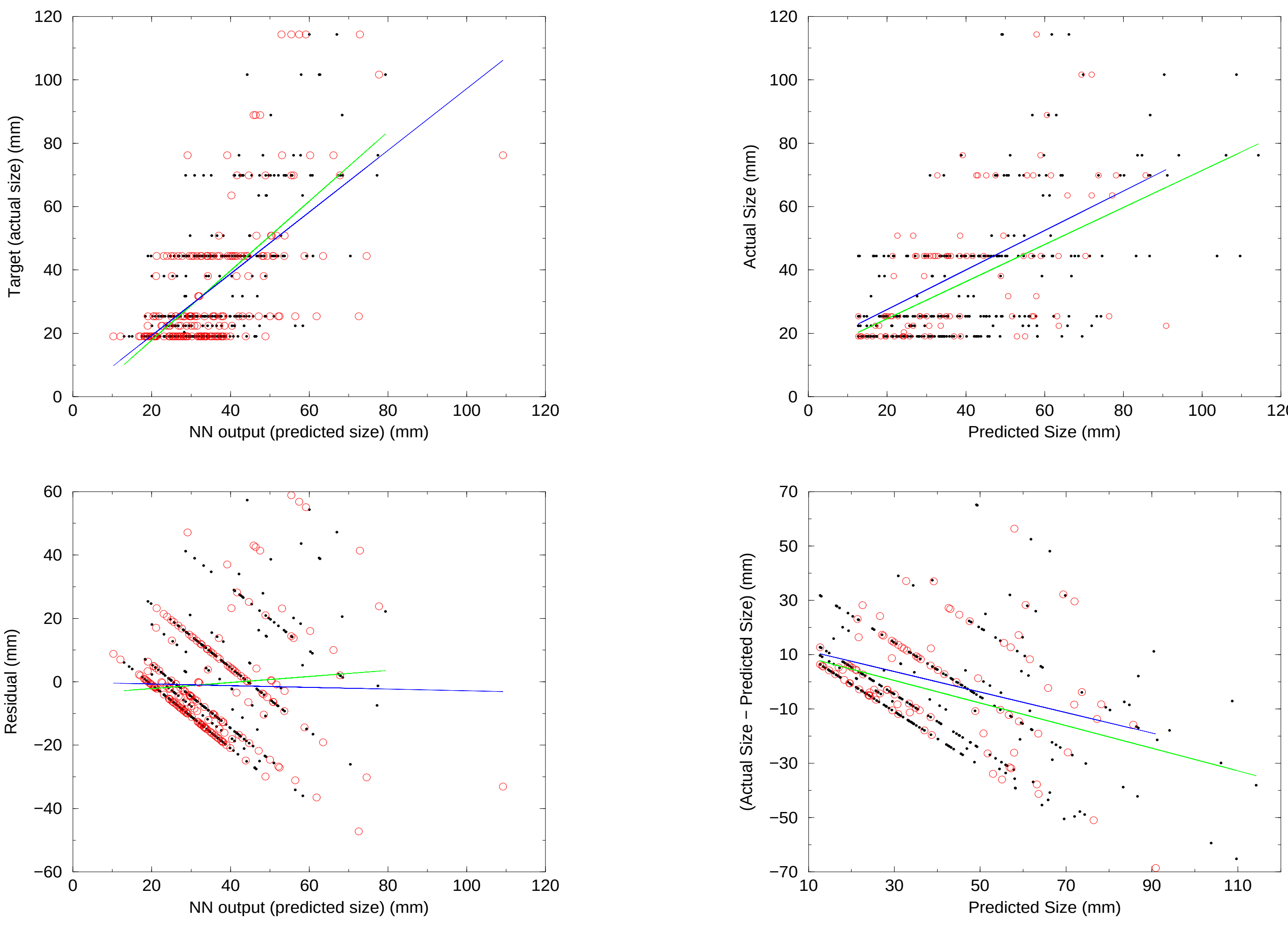


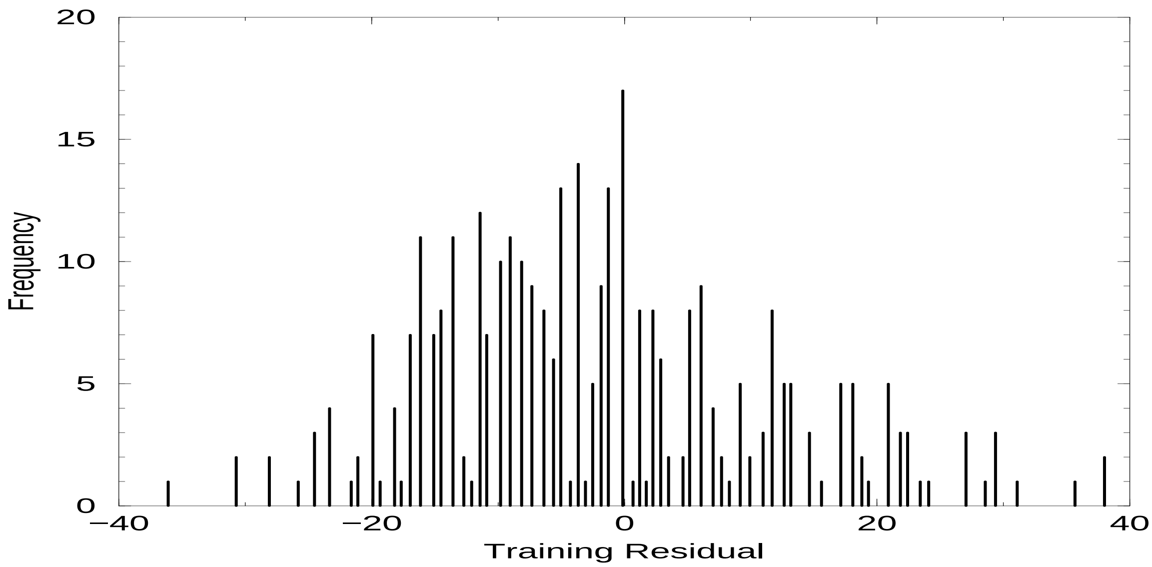
Look at r, but do not select inputs based on r.

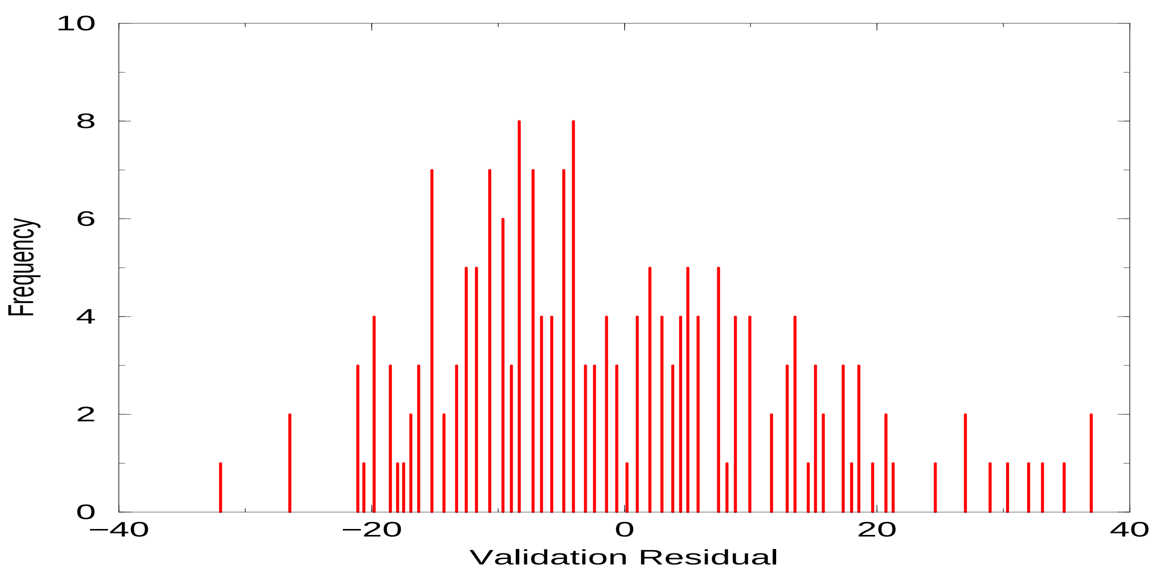
More ...

Performance






20
15
10
5
0
Frequency
−40
−20
0
20
40
Training Residual


10
8
6
4
2
0
Frequency
−40
−20
0
20
40
Validation Residual

Discussion

- The NN outperforms "Rule".
- Having said that, avoid model comparison!
- Use graphical (rather than scalar) means of assessing performance (scatterplots, residual plots, etc.)
- The linear correlation coefficient of the scatterplot, r, is a good scalar measure of performance. r^2 is the percentage of total variance explained. $r \sim 1$ is good.
- Any nonlinear pattern in residual plots suggests that the NN has learned the wrong function.
- The r of the residual plots is a good scalar measure of performance. $r \sim 1$ is good.
- $\exists$ many corrections to these r's accounting for nonlinearity, no. of weights, etc.
- Don't diagnose the weights.
- "Curse of dimensionality".

Single Lesson

NNs do statistics. So, start from linear regression.