The University Of Sheffield.

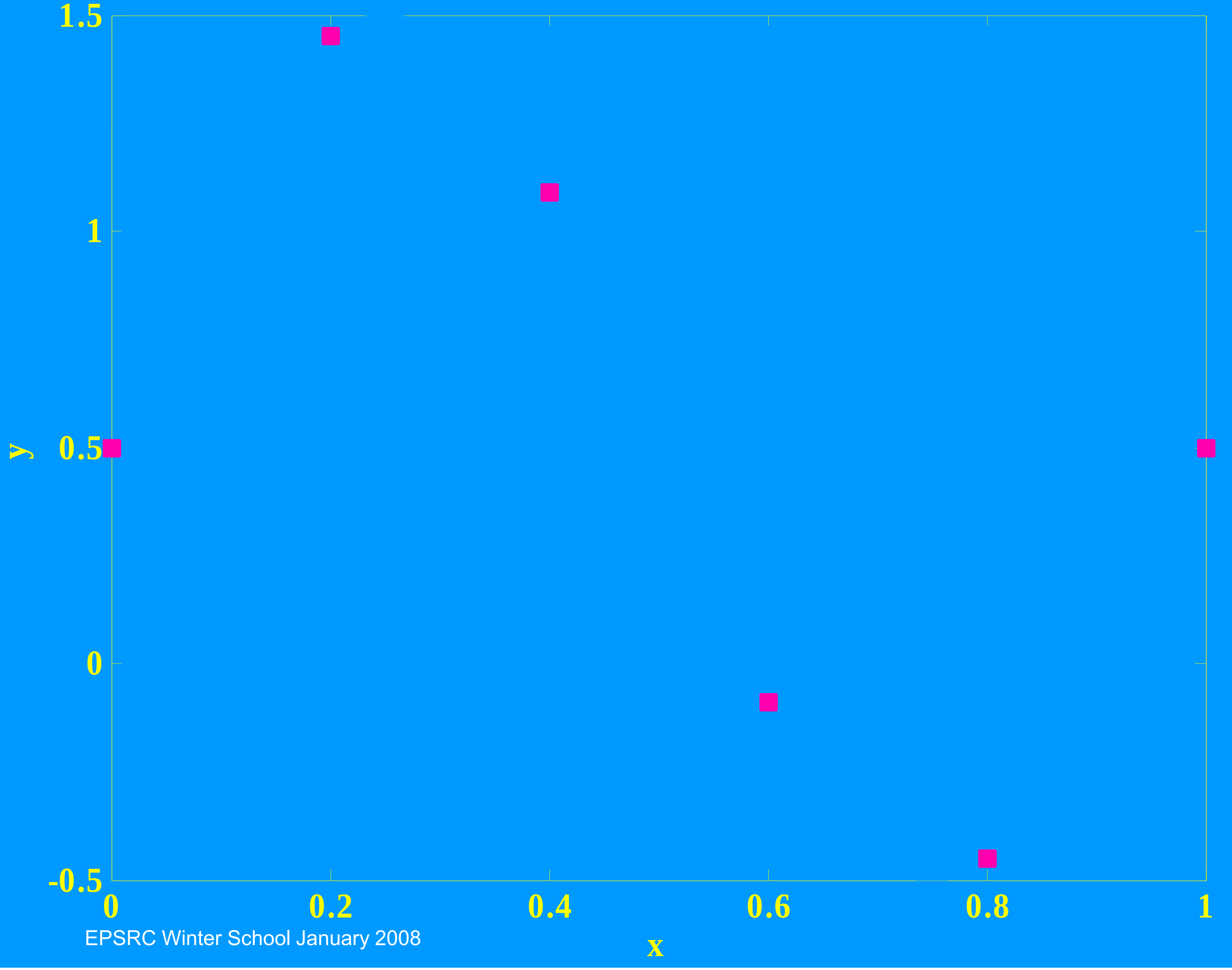
Non-linear Modelling by Adaptive Pre-processing

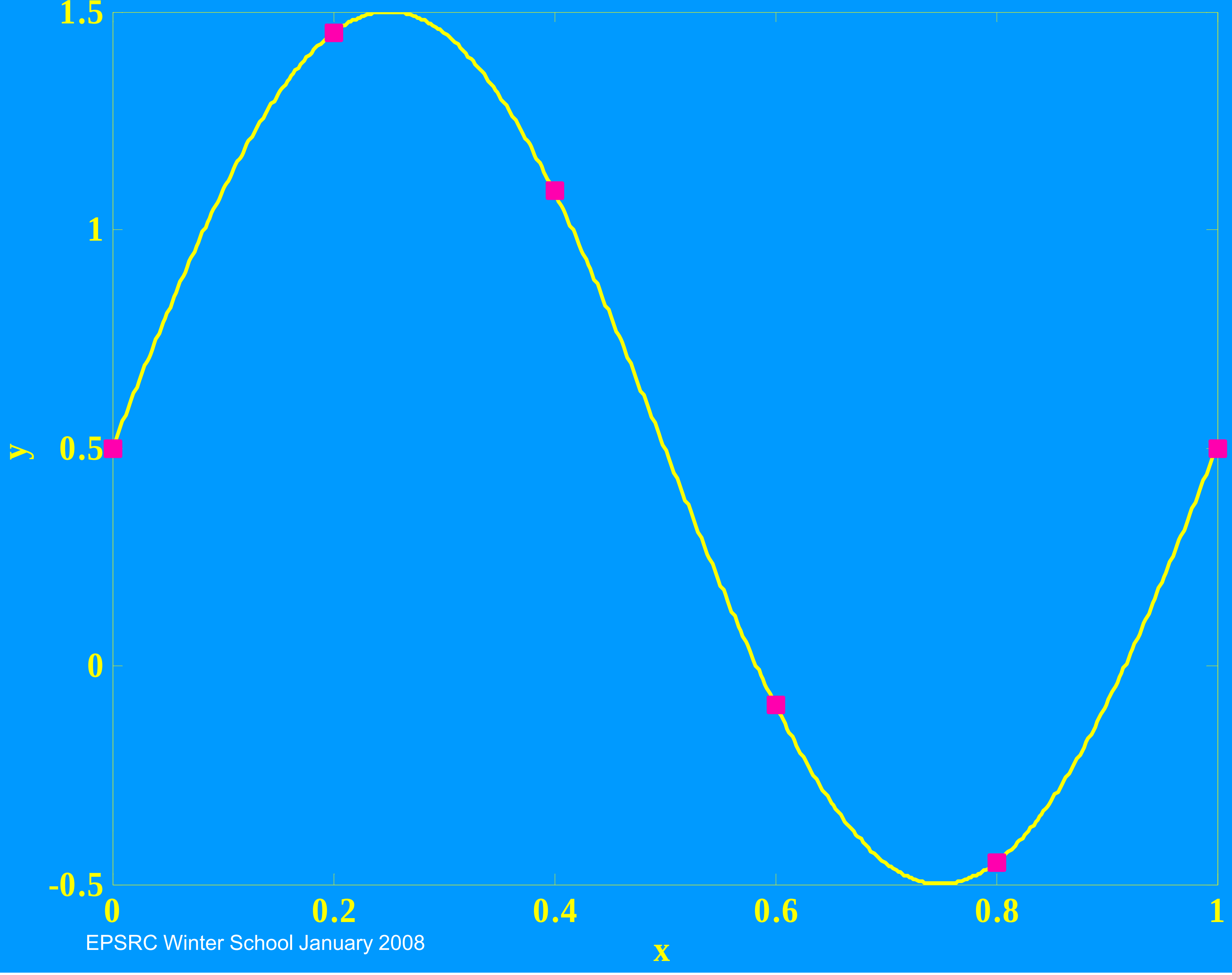
Rob Harrison
Automatic Control & Systems Engineering

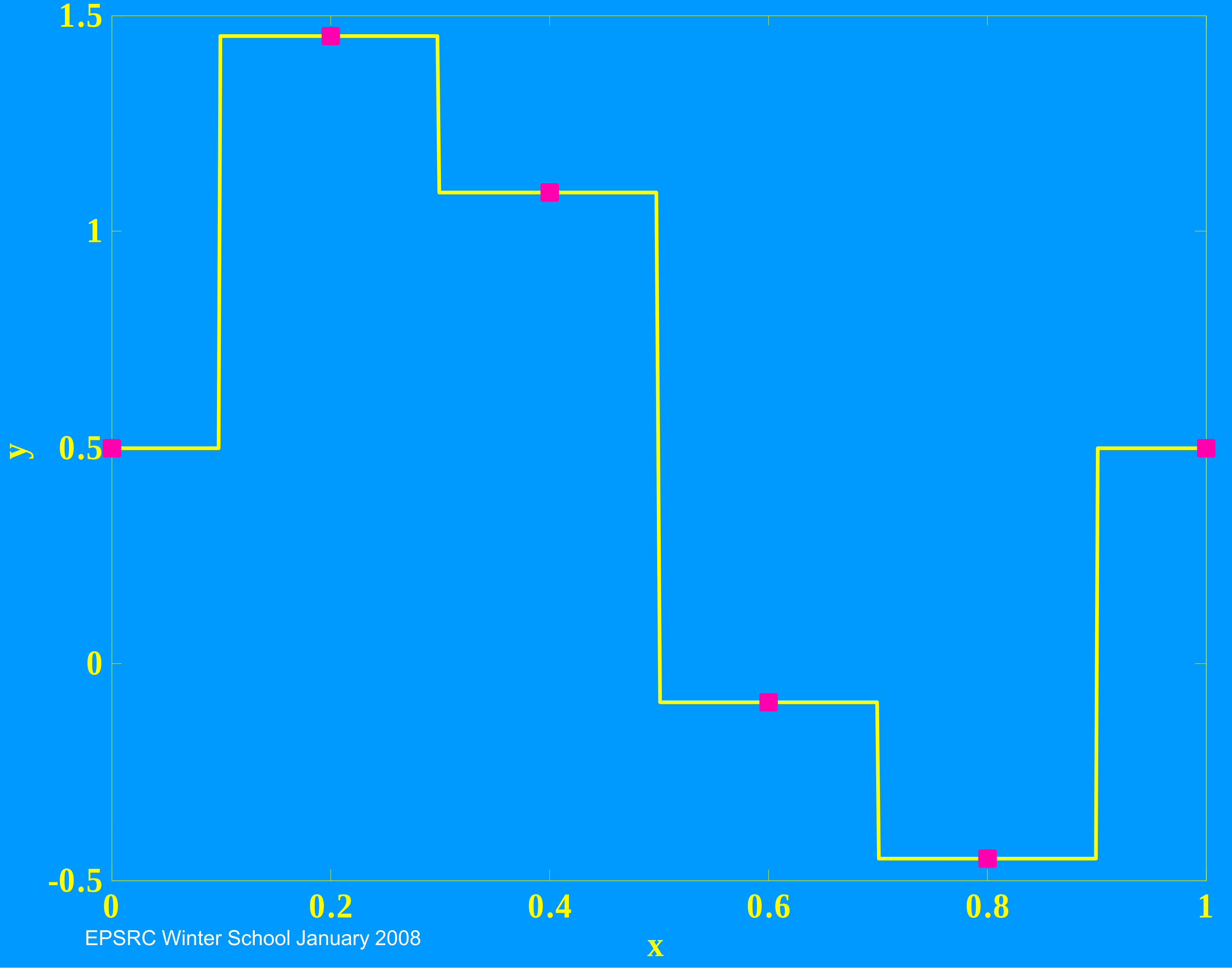
The University Of Sheffield.

The Data Modelling Problem

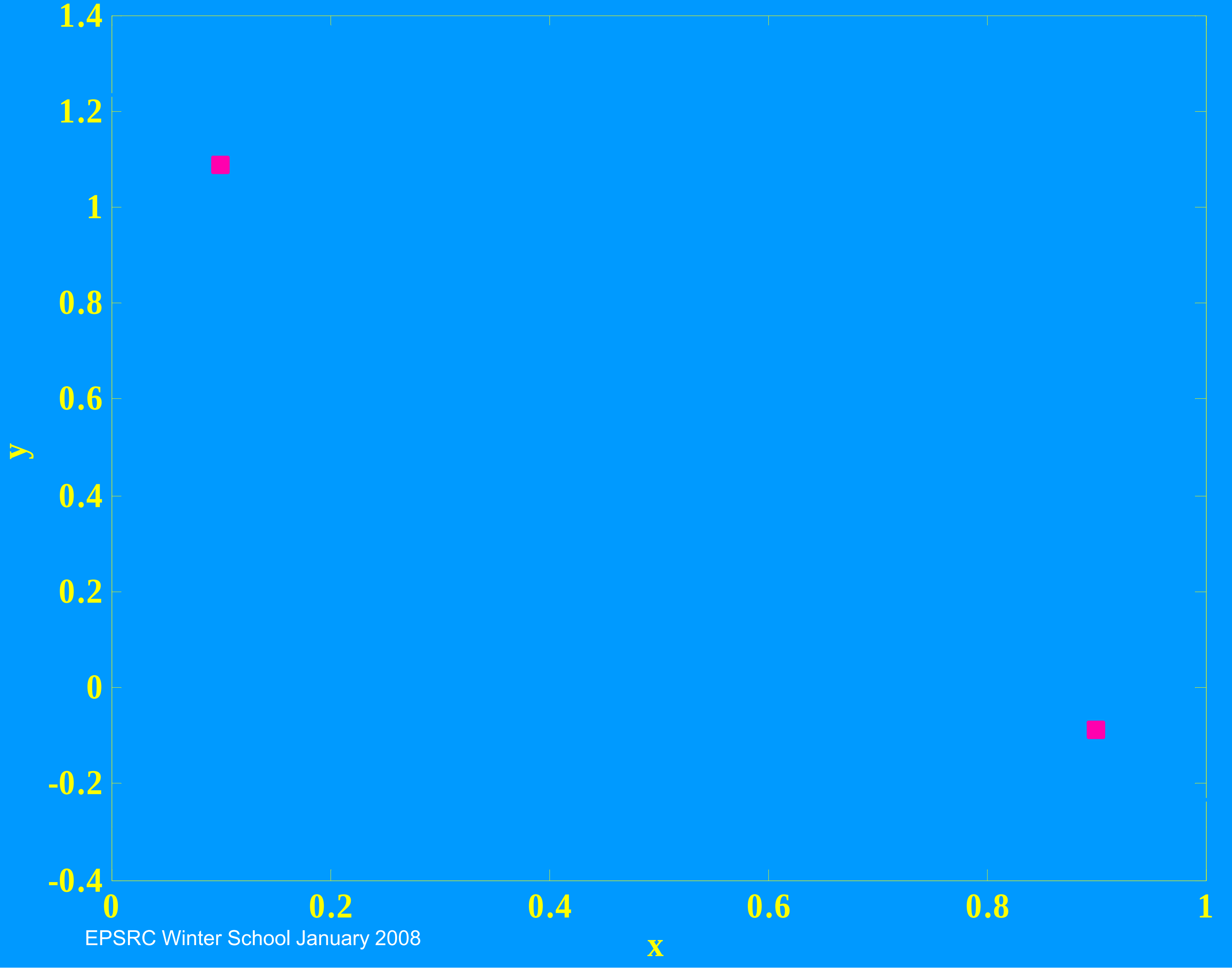
- $y= f(x)$ $z=y+e$
- multivariate & non-linear
  - measurement errors
- $\{x_i, z_i\}$ $i = 1:N$ $z_i = f(x_i)+e_i$
- infer behaviour *everywhere* from a few examples
  - little or no prior information on $f(x)$
- $\hat{y}$ etc. indicates *estimate*


1.5
1
0.5
0
-0.5
y
0
0.2
0.4
0.6
0.8
1
x

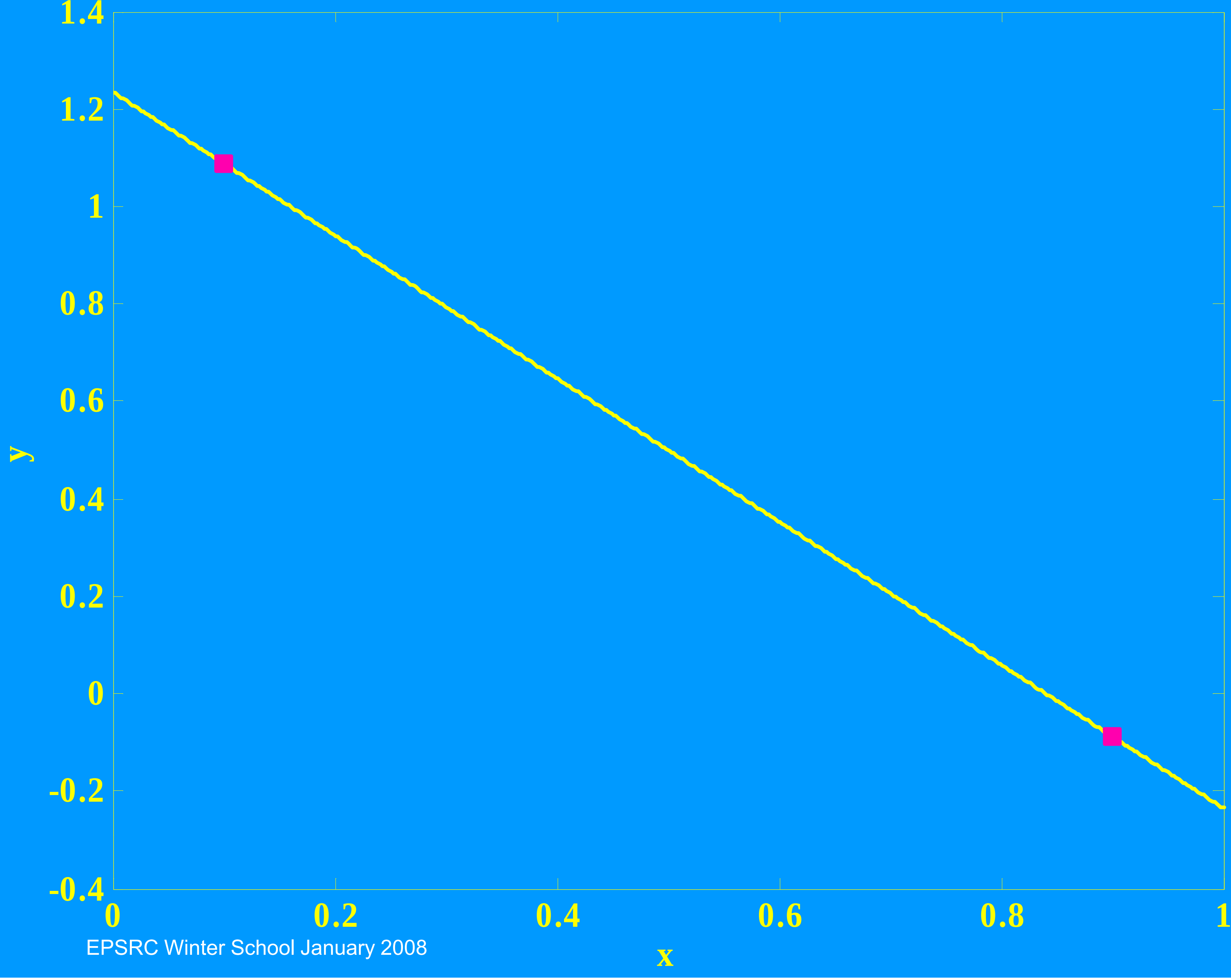

1.5
1
0.5
0
-0.5
y
0
0.2
0.4
0.6
0.8
1
x

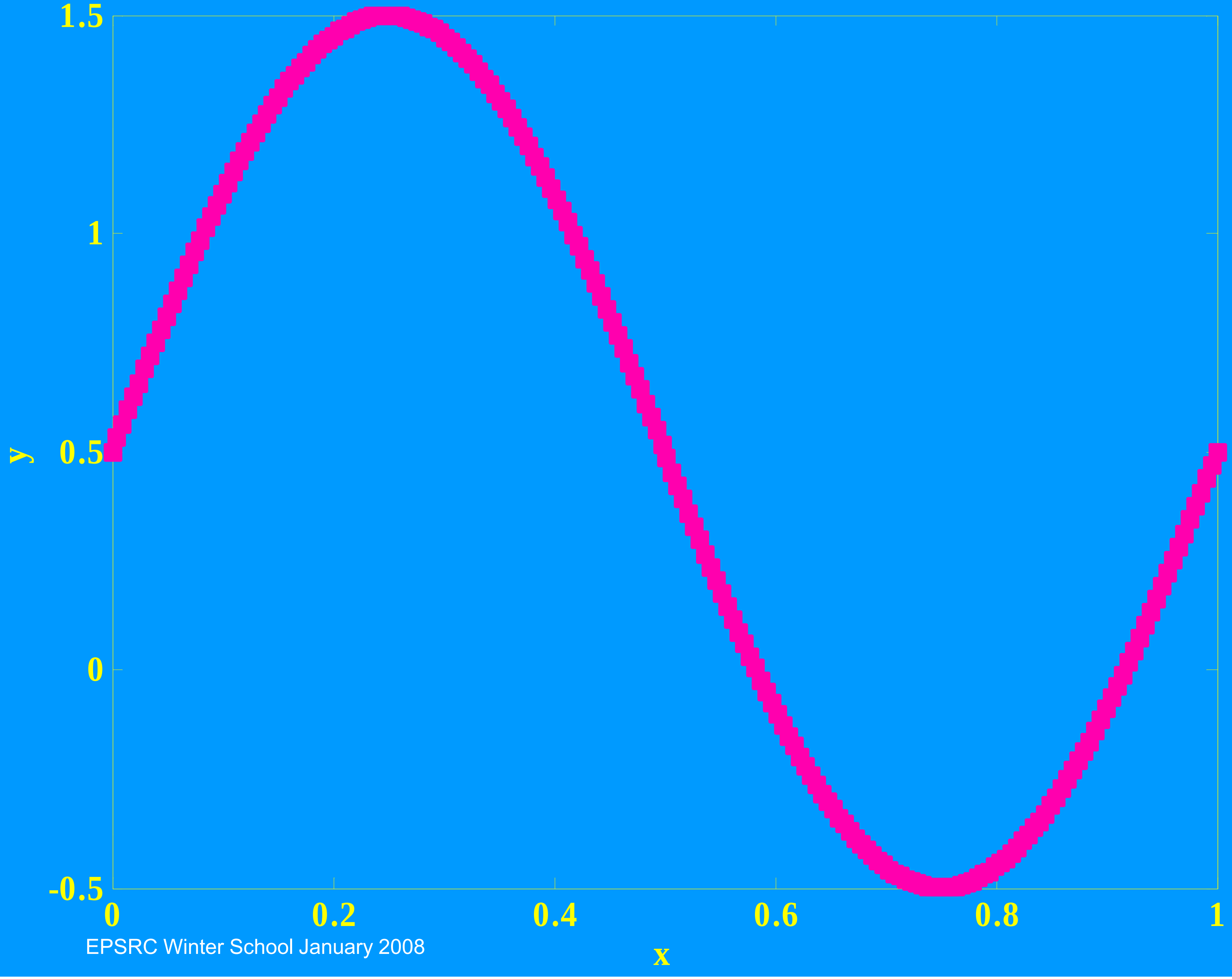


1.5
1
0.5
0
-0.5
y
0
0.2
0.4
0.6
0.8
1
x



1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
y
0
0.2
0.4
0.6
0.8
1
x


1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
y
0
0.2
0.4
0.6
0.8
1
x


1.5
1
0.5
0
-0.5
y
0
0.2
0.4
0.6
0.8
1
x

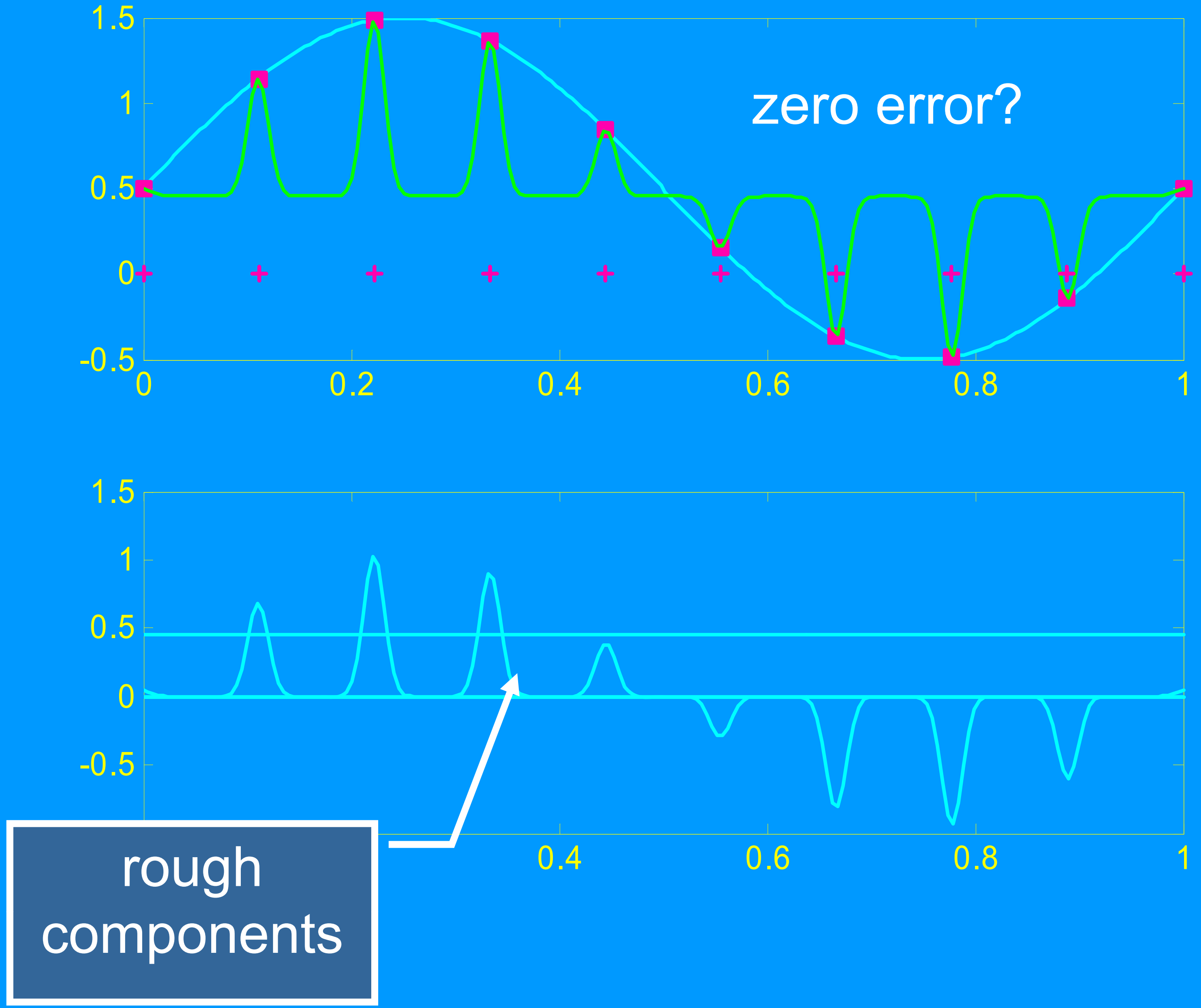
Dimensionality

- lose ability to *see* the shape of f(x)
  - try it in 13-D
- number of samples exponential in d
  - if N OK in 1-D, N^d needed in d-D
- how do we know if “well-spaced”?
  - how can we sample where the action is?
  - observational vs experimental data!
- ALWAYS undersampled!

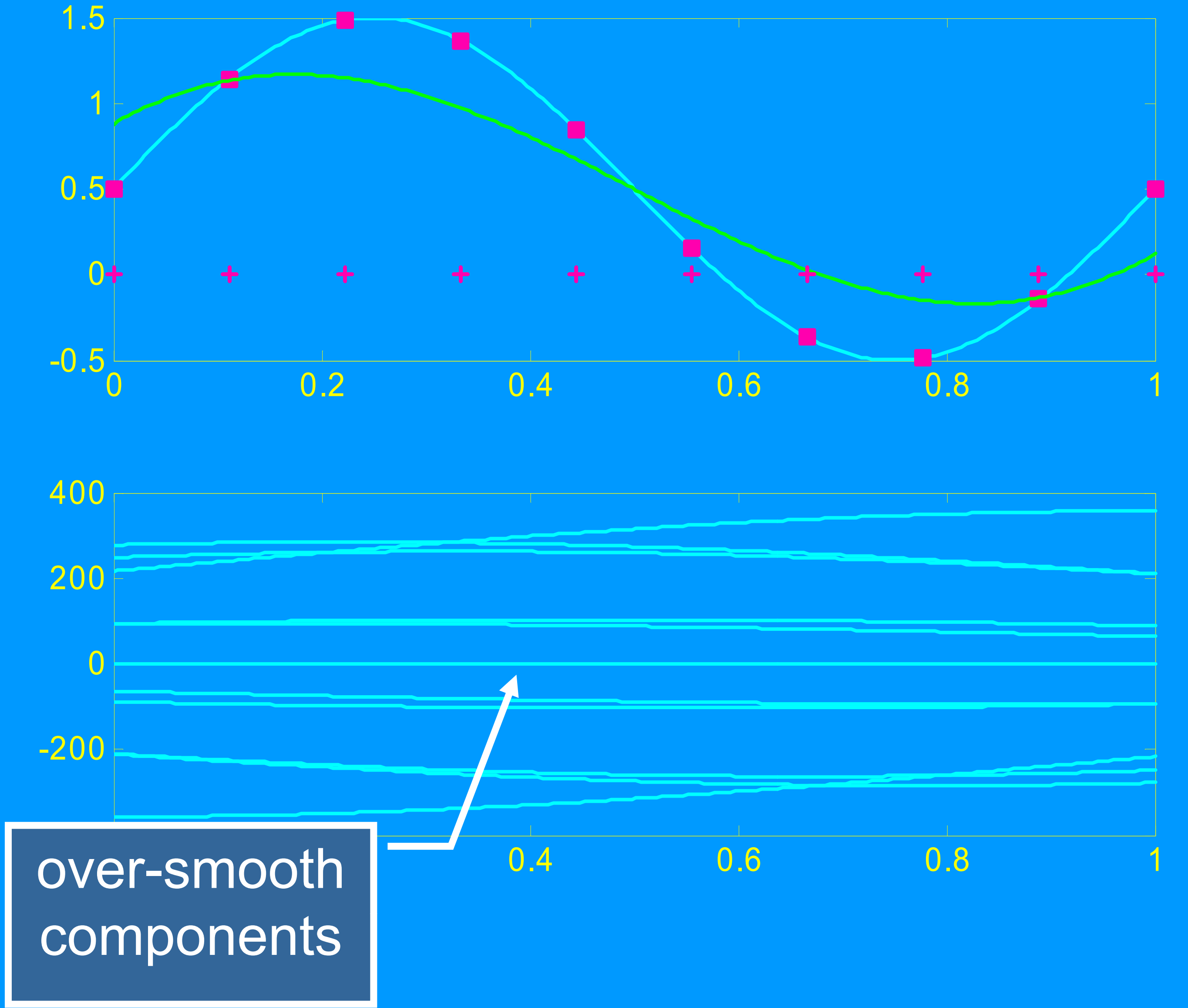
What goes on in the Gaps?

- Universal Approximation
- Advantage
  - can bend to (almost) any shape
- Disadvantage
  - can bend to (almost) any shape
- Training data is all we have to go on

Overfitting (sample data)



Underfitting (sample data)



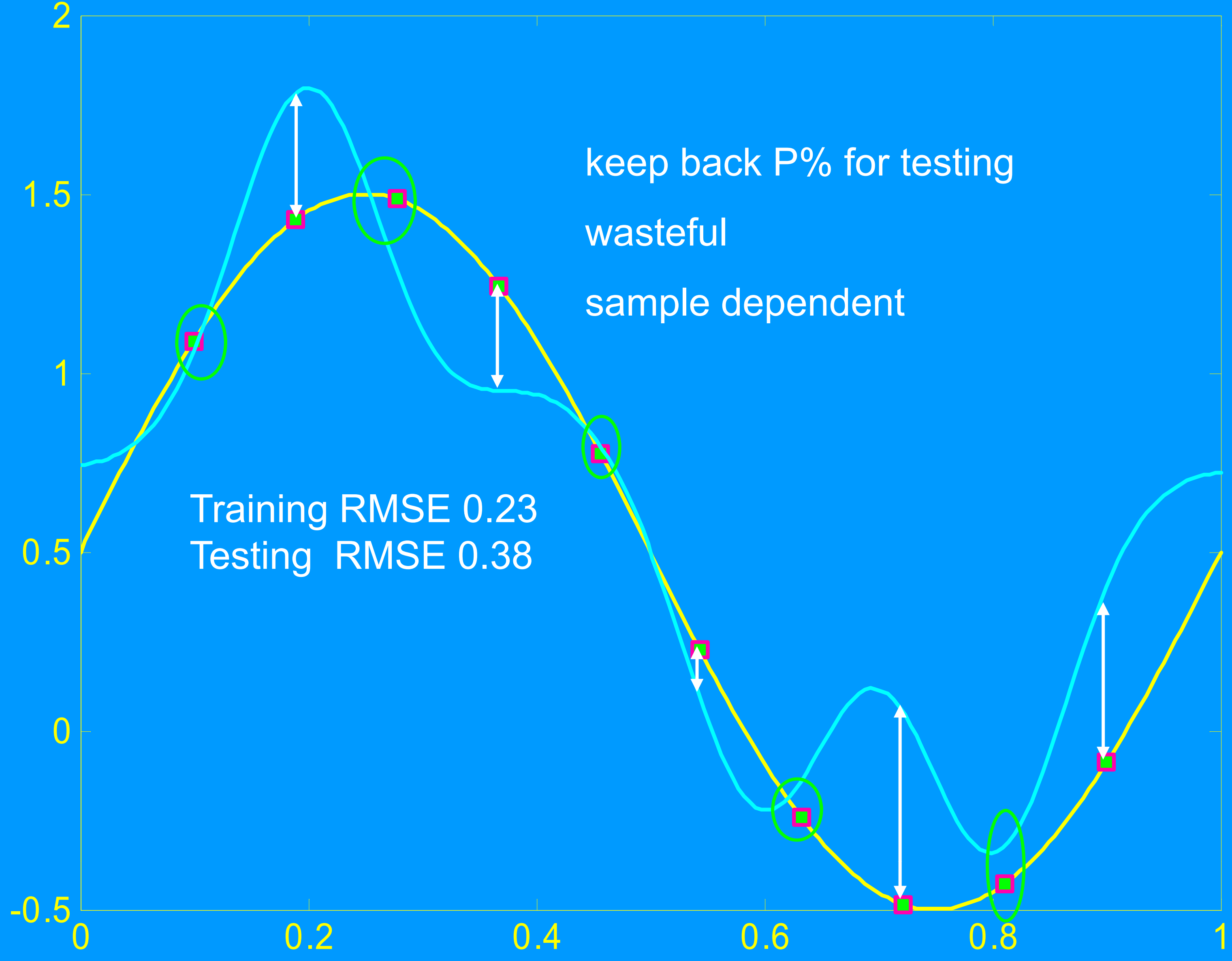
Goldilocks

1.5
1
0.5
0
-0.5
0
0.2
0.4
0.6
0.8
1
v. small error
1
0.5
0
-0.5
0.4
0.6
0.8
1
“just right”
components

Restricting "Flexibility"

- use data to tell the estimator how to behave
- regularization/penalization
  - penalize "roughness"
  - e.g. SSE + ρQ
  - $Q = \Sigma w^2_{ij} \rightarrow \hat{w} = (\Phi^T\Phi + \rho I)^{-1} \Phi^T z$
- use potentially complex structure
  - data constrains where it can
  - Q constrains elsewhere

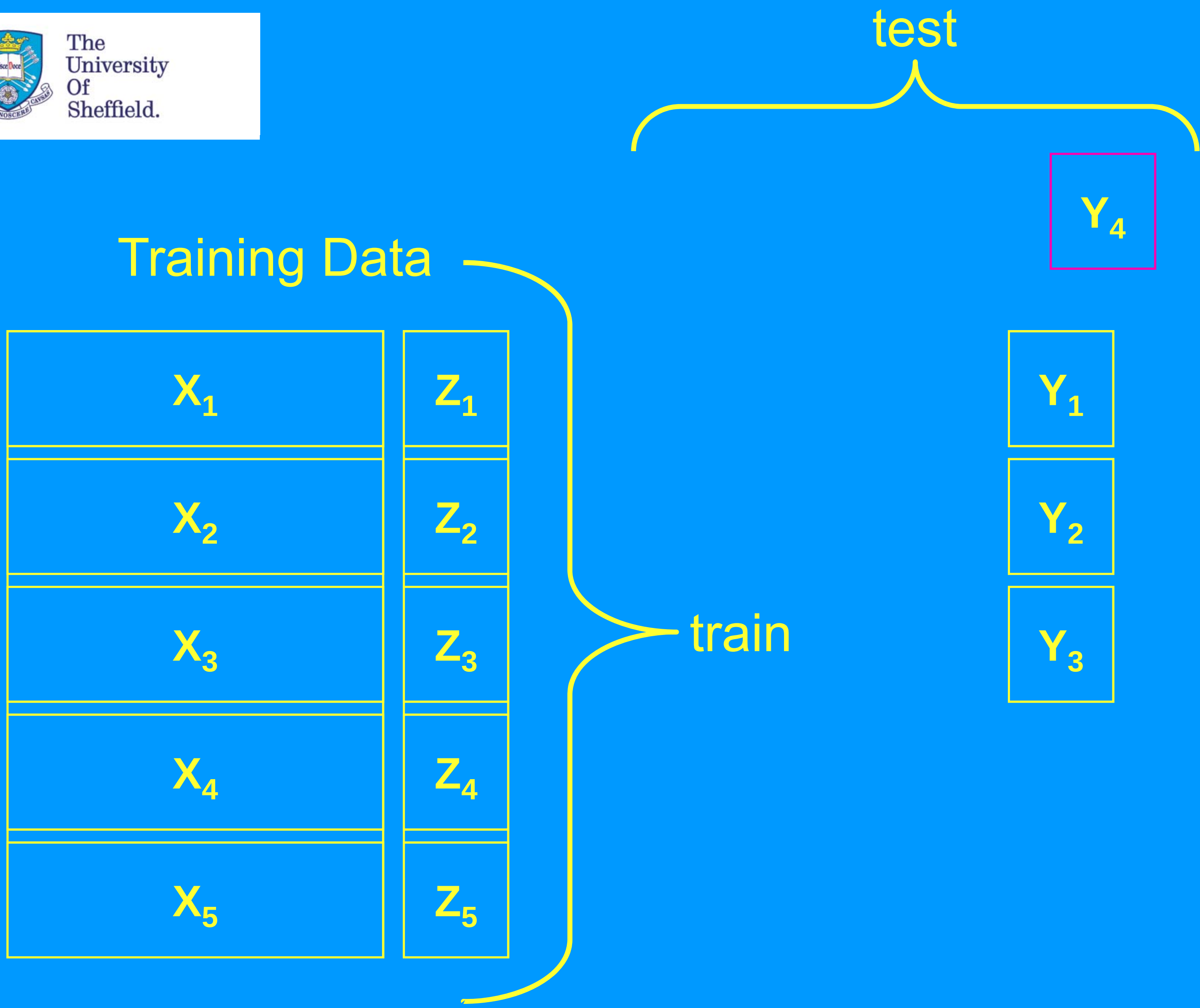
Hold-out Method

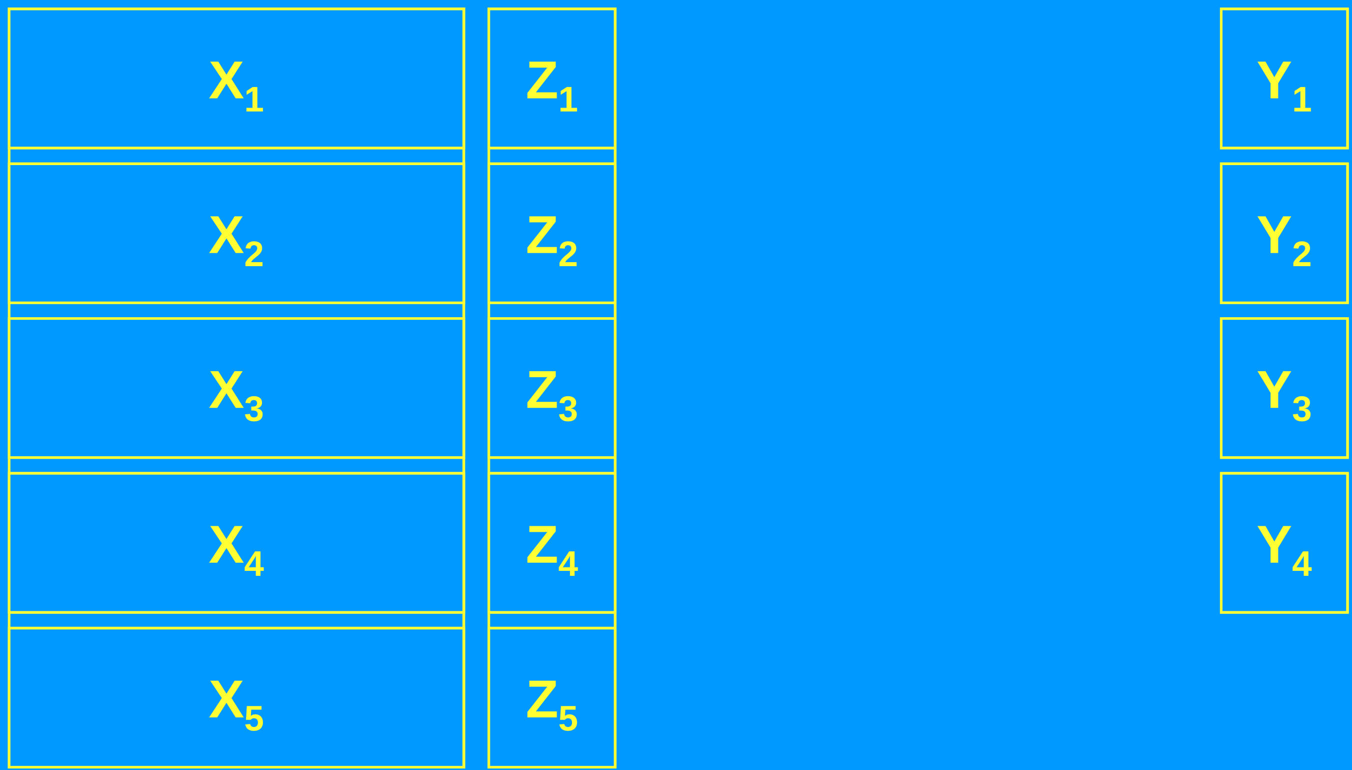


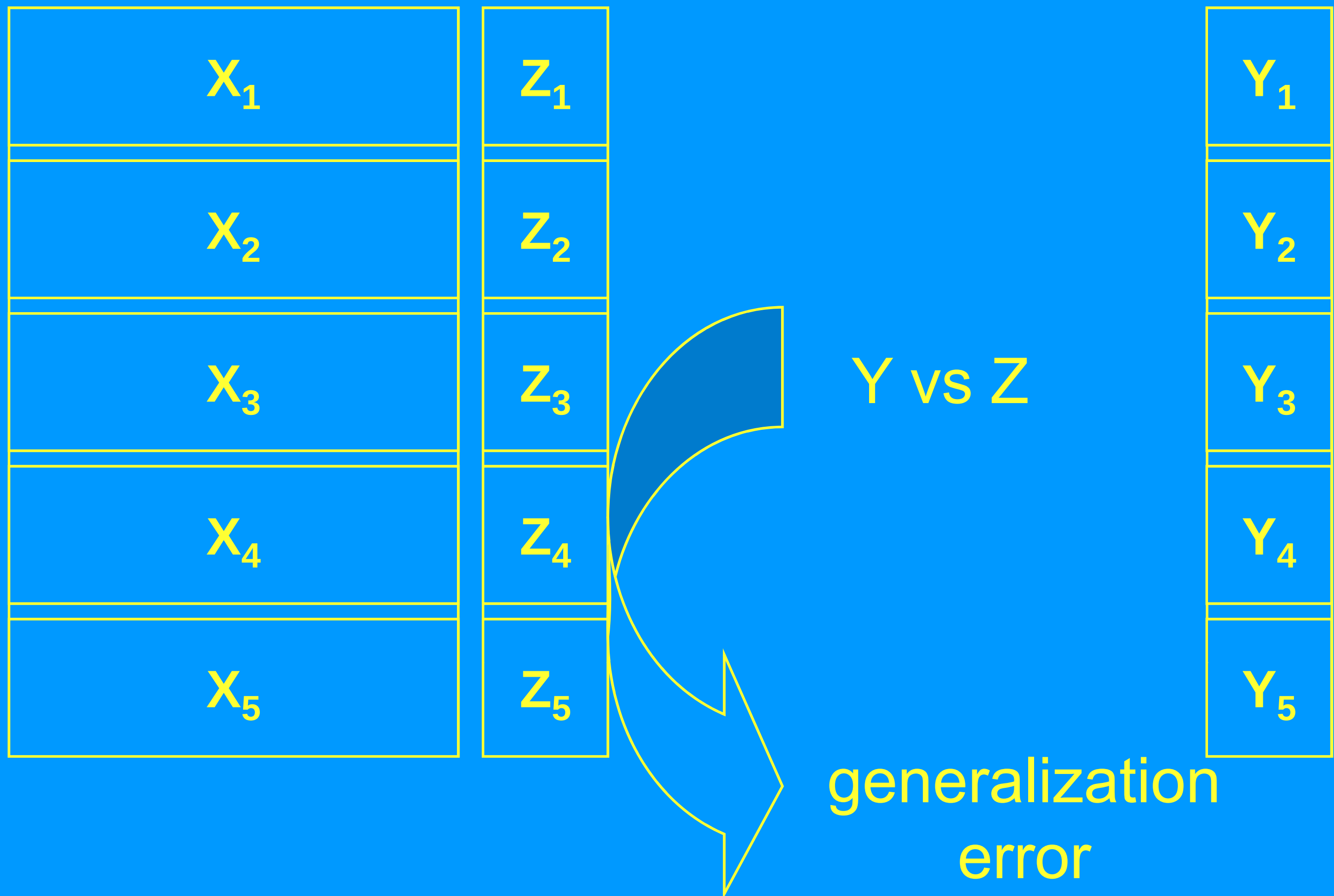
The University Of Sheffield.

Cross Validation

- leave-one-out CV
  - train on all but one
  - test that one
  - repeat N times
  - compute performance
- m-fold CV
  - divide sample into m non-overlapping sets
  - proceed as above
- all data used for training and testing
  - more work but realistic performance estimates
- used to choose “hyper-parameters”
  - e.g. ρ, number, width …


test
Y4
Training Data
X1
X2
X3
X4
X5
Z1
Z2
Z3
Z4
Z5
train
Y1
Y2
Y3


X_1
X_2
X_3
X_4
X_5
Z_1
Z_2
Z_3
Z_4
Z_5
Y_1
Y_2
Y_3
Y_4



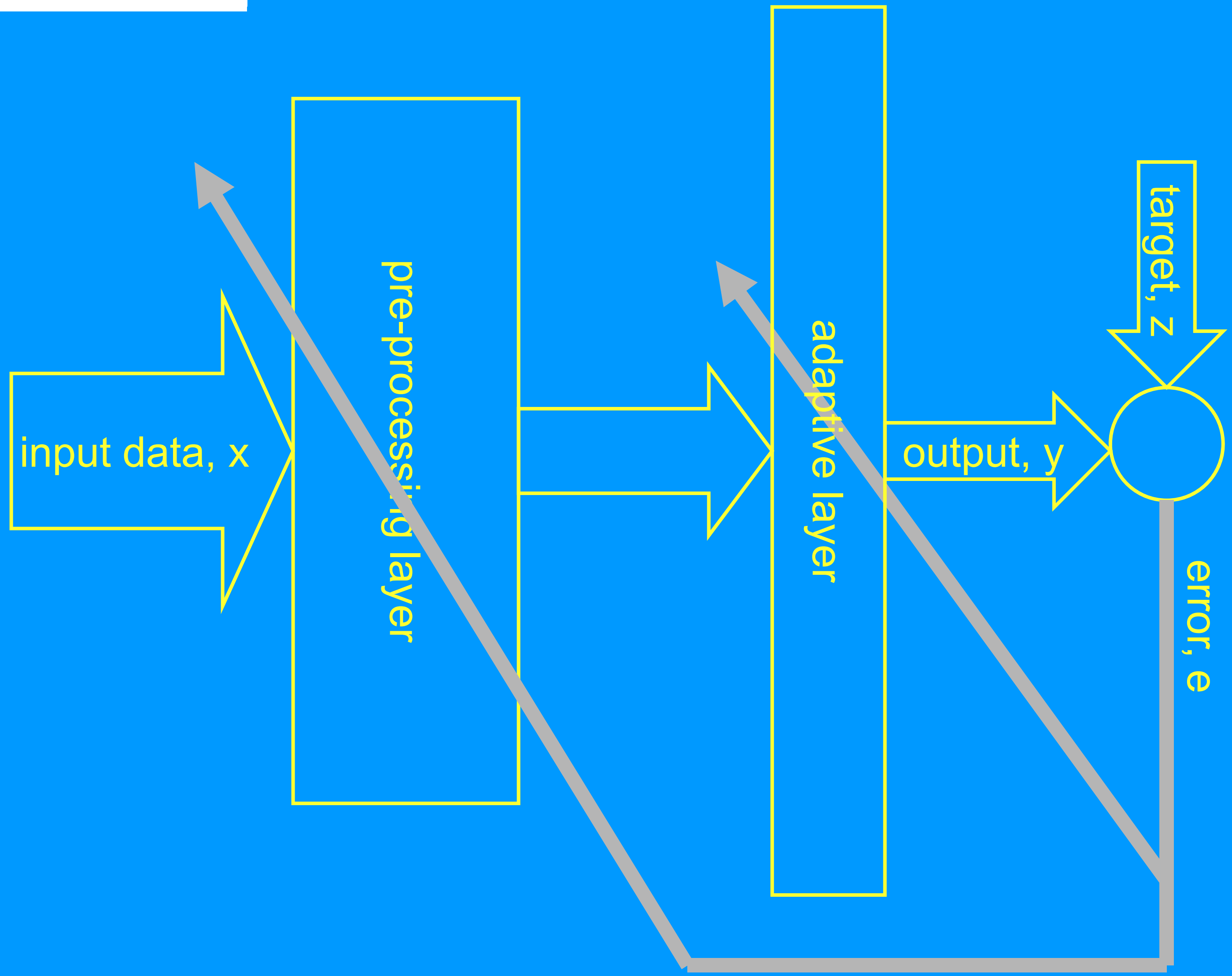
X_1
X_2
X_3
X_4
X_5
Z_1
Z_2
Z_3
Z_4
Z_5
Y vs Z
generalization error
Y_1
Y_2
Y_3
Y_4
Y_5

Adaptive Basis Functions

- "linear" models
  - fixed pre-processing
  - parameters → cost "benign"
  - easy to optimize

    but
  - combinatorial
  - arbitrary choices

what is best pre-processor to choose?


input data, x
pre-processing layer
adaptive layer
output, y
target, z
error, e

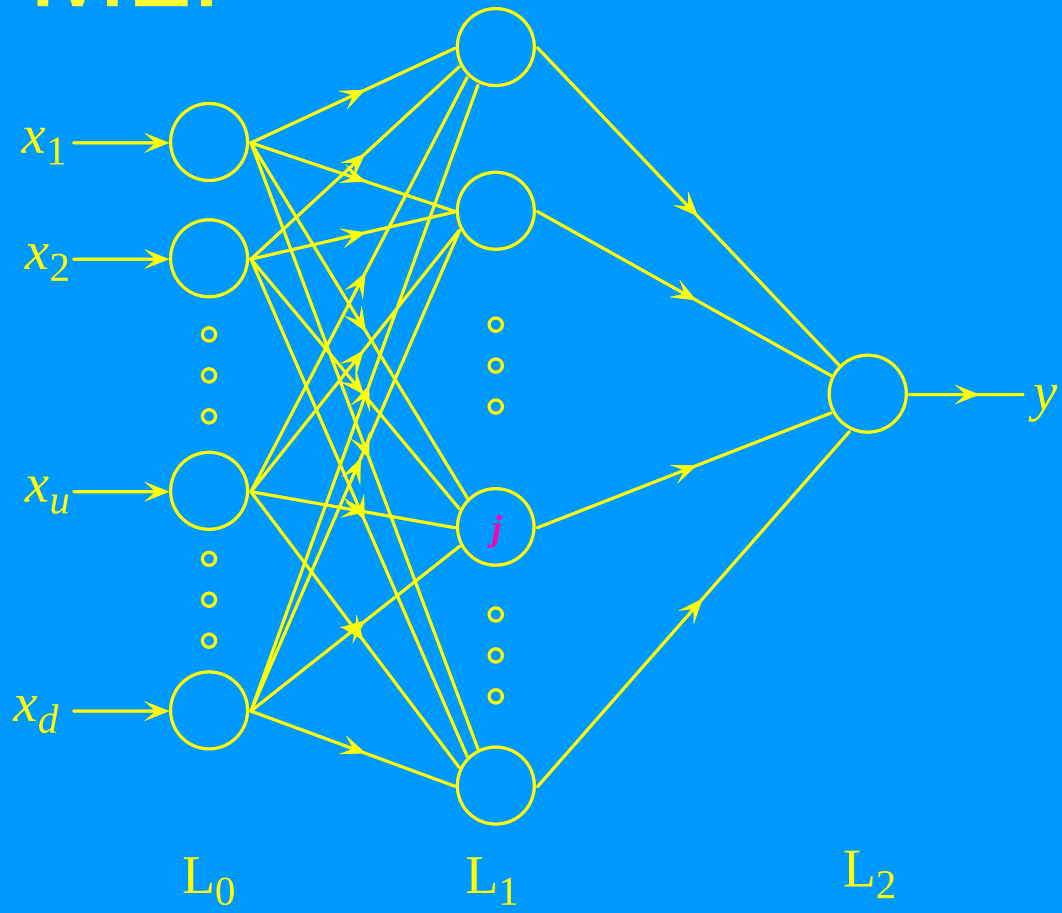
The University Of Sheffield.

The Multi-Layer Perceptron

- formulated from loose biological principles
- popularized mid 1980s
  - Rumelhart, Hinton & Williams 1986
    Werbos 1974, Ho 1964
- "learn" pre-processing stage from data
- layered, feed-forward structure
  - sigmoidal pre-processing
  - task-specific output

non-linear model

Two-Layer MLP



$$y = \theta\left(\underline{w}^T \underline{v} + b\right)$$

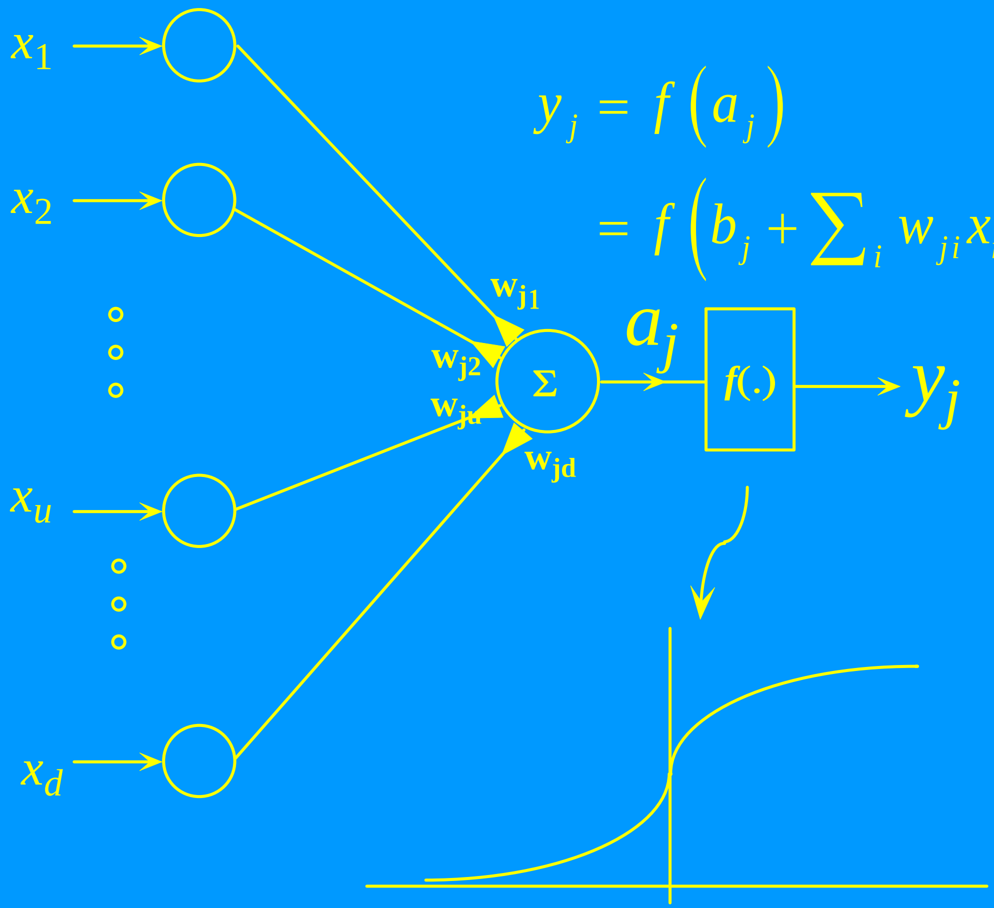
$$v_j = \sigma\left(\underline{w}_j^T \underline{x} + b_j\right)$$

$$y = \theta\left(\sum_i w_i \sigma\left(\sum_j w_j x_j + b_j\right) + b\right)$$

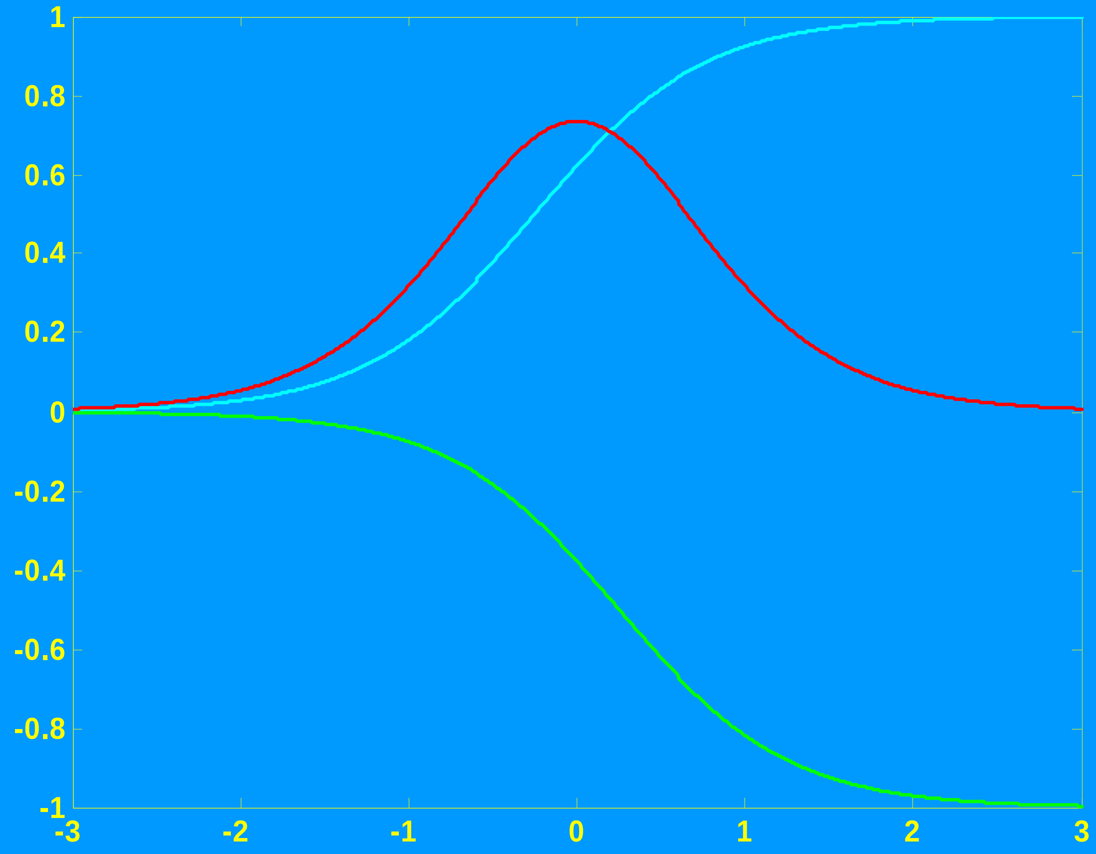
The University Of Sheffield.

A Sigmoidal Unit

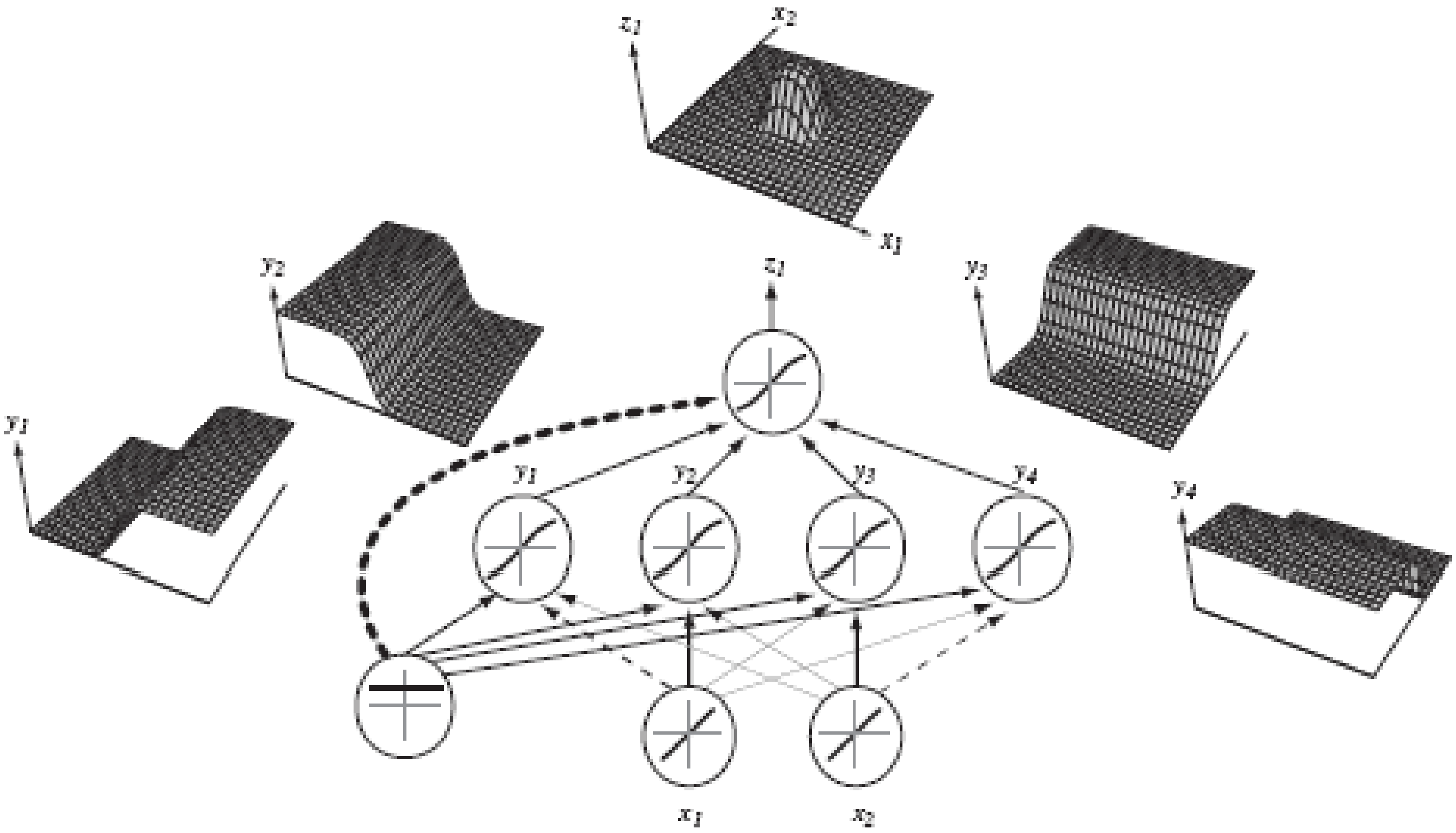
$$y_j = f(a_j) = f\left(b_j + \sum_i w_{ji} x_i\right)$$



Combinations of Sigmoids



The University Of Sheffield.



Universal Approximation

- linear combination of “enough” sigmoids
  - Cybenko, 1990
- single hidden layer adequate
  - more may be better
- choose hidden parameters (w, b) optimally

problem solved?

Interpretation

- Minimising SSE equivalent to finding *conditional mean* of target data
  - infinite sample / global minimum

MLP

$$J_{\infty} = \frac{1}{2}\sum_{i=1}^{n}\int_{\underline{x}\in\mathbb{R}^d}\left(E\left[z_i \mid \underline{x}\right] - \hat{f}_i\left(\underline{x};\mathscr{W}\right)\right)^2 p\left(\underline{x}\right)d\underline{x}$$

$$+\frac{1}{2}\sum_{i=1}^{n}\int_{\underline{x}\in\mathbb{R}^d}\left(z_i - E\left[z_i \mid \underline{x}\right]\right)^2 p\left(\underline{x}\right)d\underline{x}$$

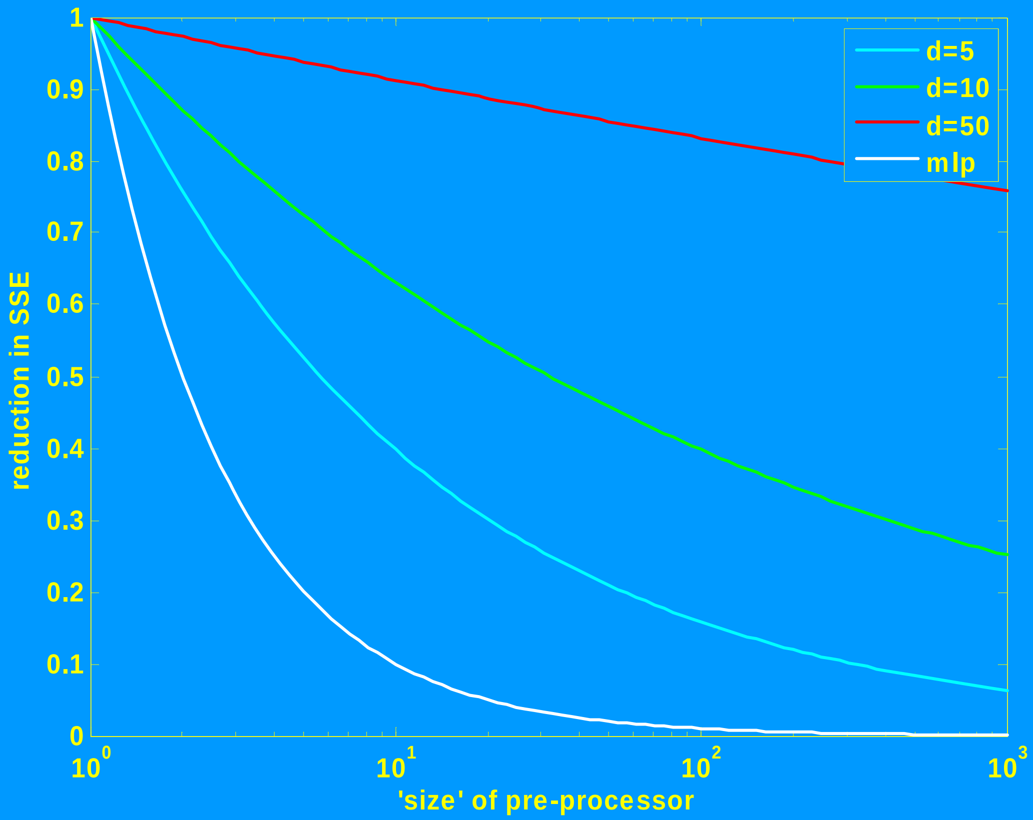
does not depend on $\mathscr{W}$

$$\boxed{\hat{f}_i\left(\underline{x};\mathscr{W}^*\right) \approx E\left[z_i \mid \underline{x}\right]}$$

Pros

- compactness
  - potential to obtain same veracity with much smaller model
  - c.f. sparsity/complexity control in linear models
- “simple” training algorithm

Compactness of Model



$$\text{MLP} \quad O\left(1/H\right)$$

$$\text{SER} \quad O\left(1/H^{2/d}\right)$$

Backpropagation Algorithm

Gradient Descent

$$w_{jr}(t+1) = w_{jr}(t) + \eta(t)\delta_j(t)y_r(t) \quad j \in L_m, r \in L_{m-1}$$ Update Rule

Generalised Delta Rule

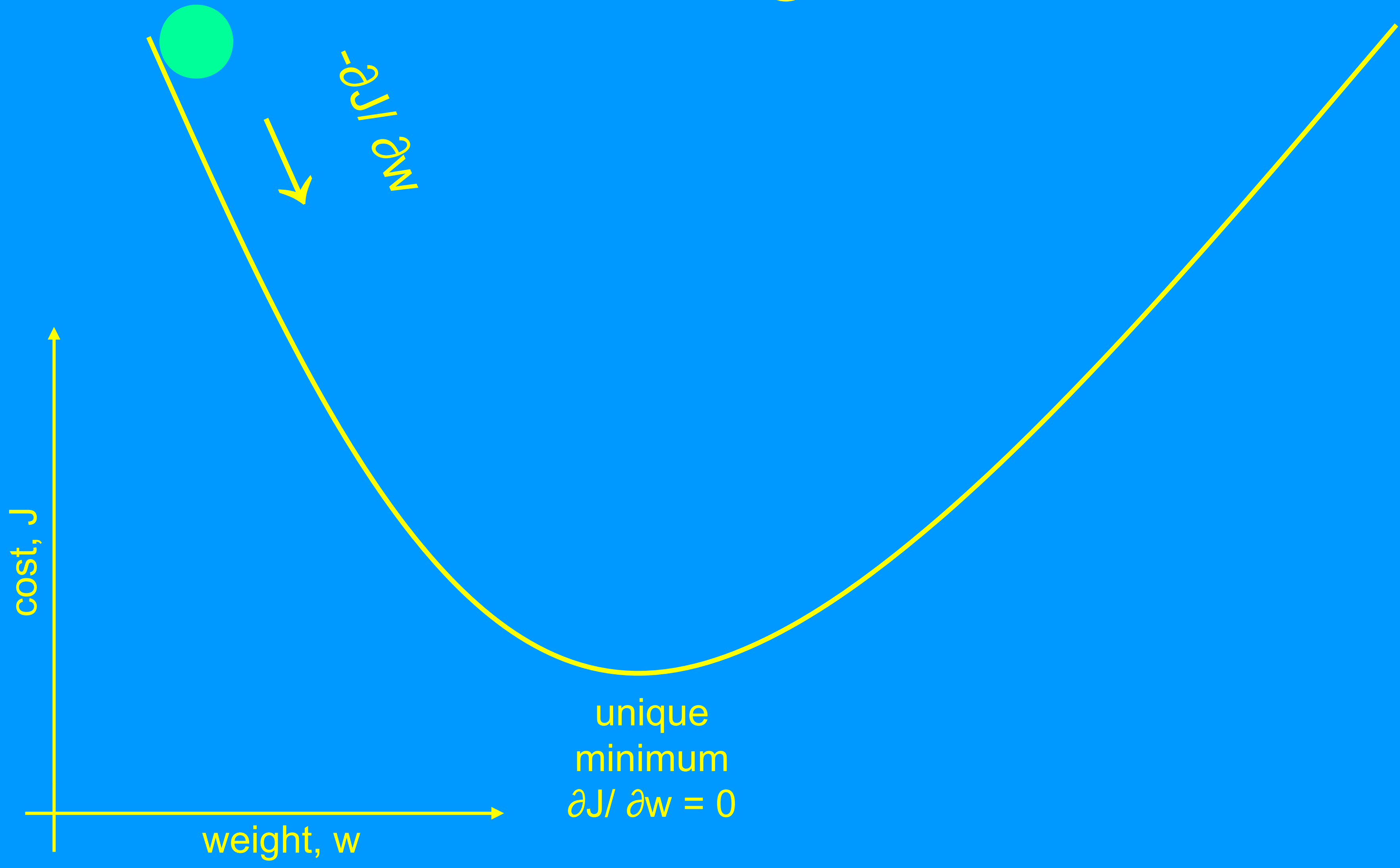
$$\delta_j(t) = \theta'\left(net_j(t)\right)e_j(t) \quad m = M$$ output layer

$$\delta_j(t) = \sigma'\left(net_j(t)\right)\sum_{i \in L_m} w_{ij}(t)\delta_i(t) \quad j \in L_{m-1}, m \in \text{hidden layers}$$

& Cons

- parameter → cost "malign"
  - optimization difficult
  - many solutions possible
  - effect of hidden weights in output non-linear

Rolling Ball

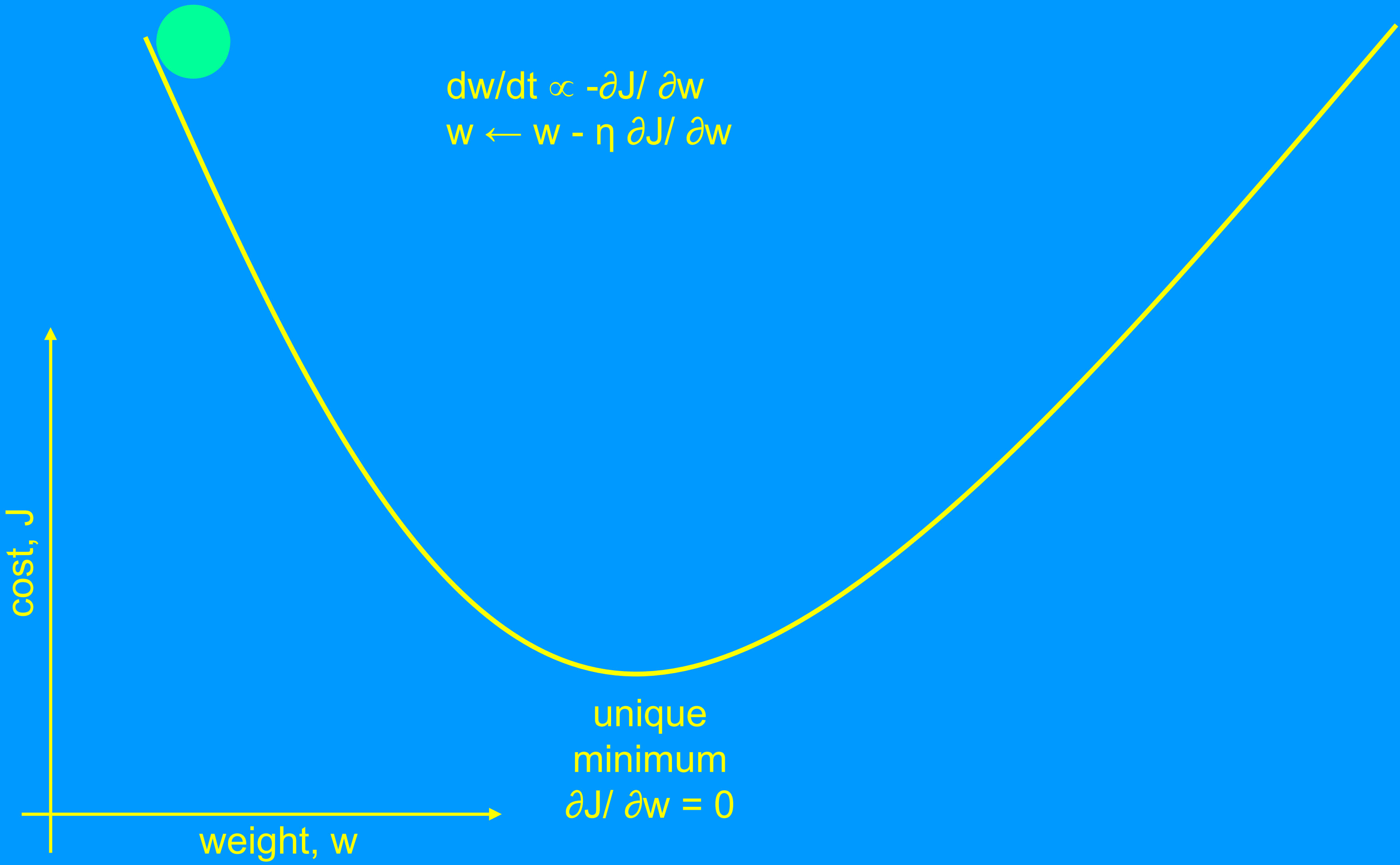


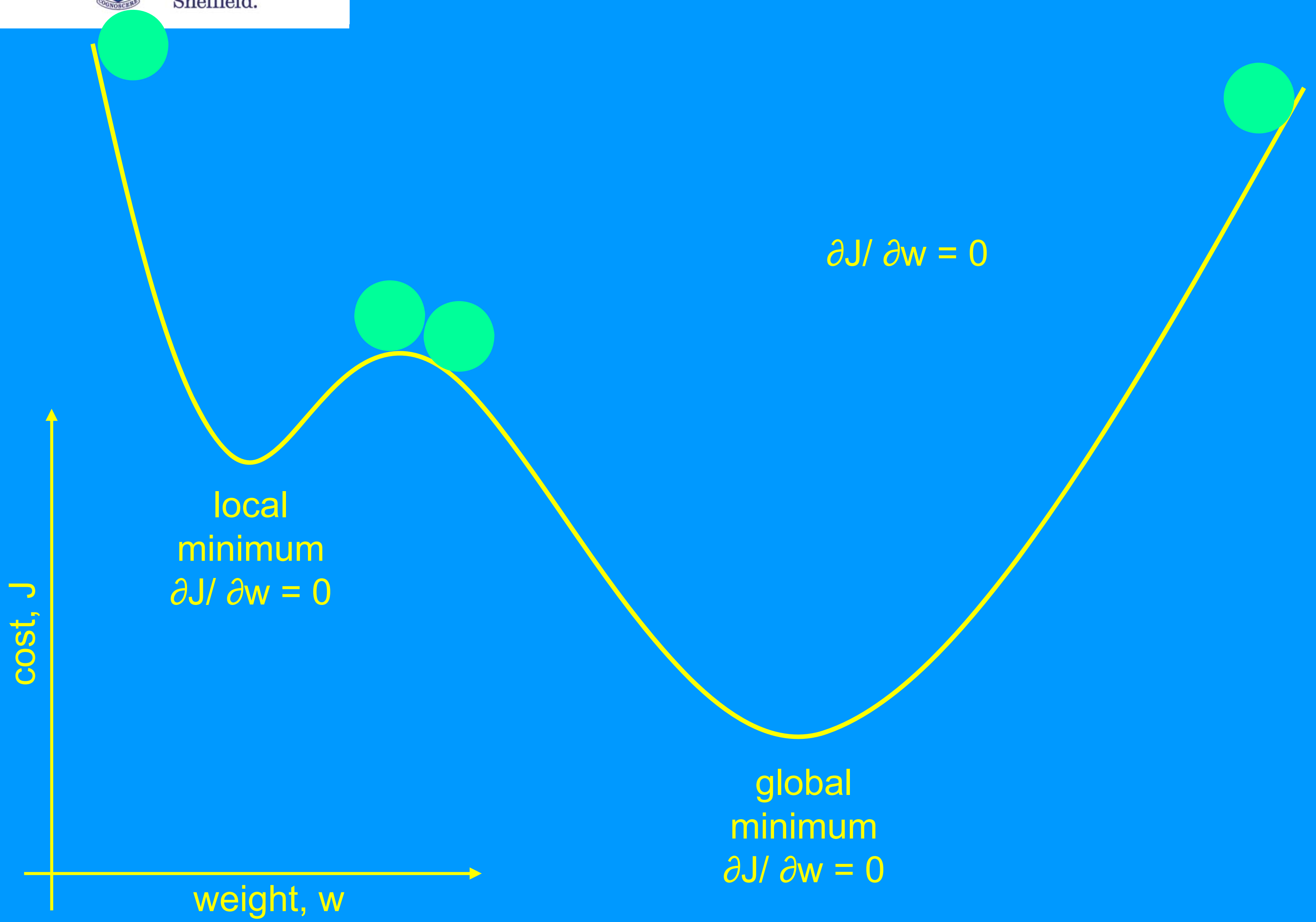
The University Of Sheffield.

Gradient Descent

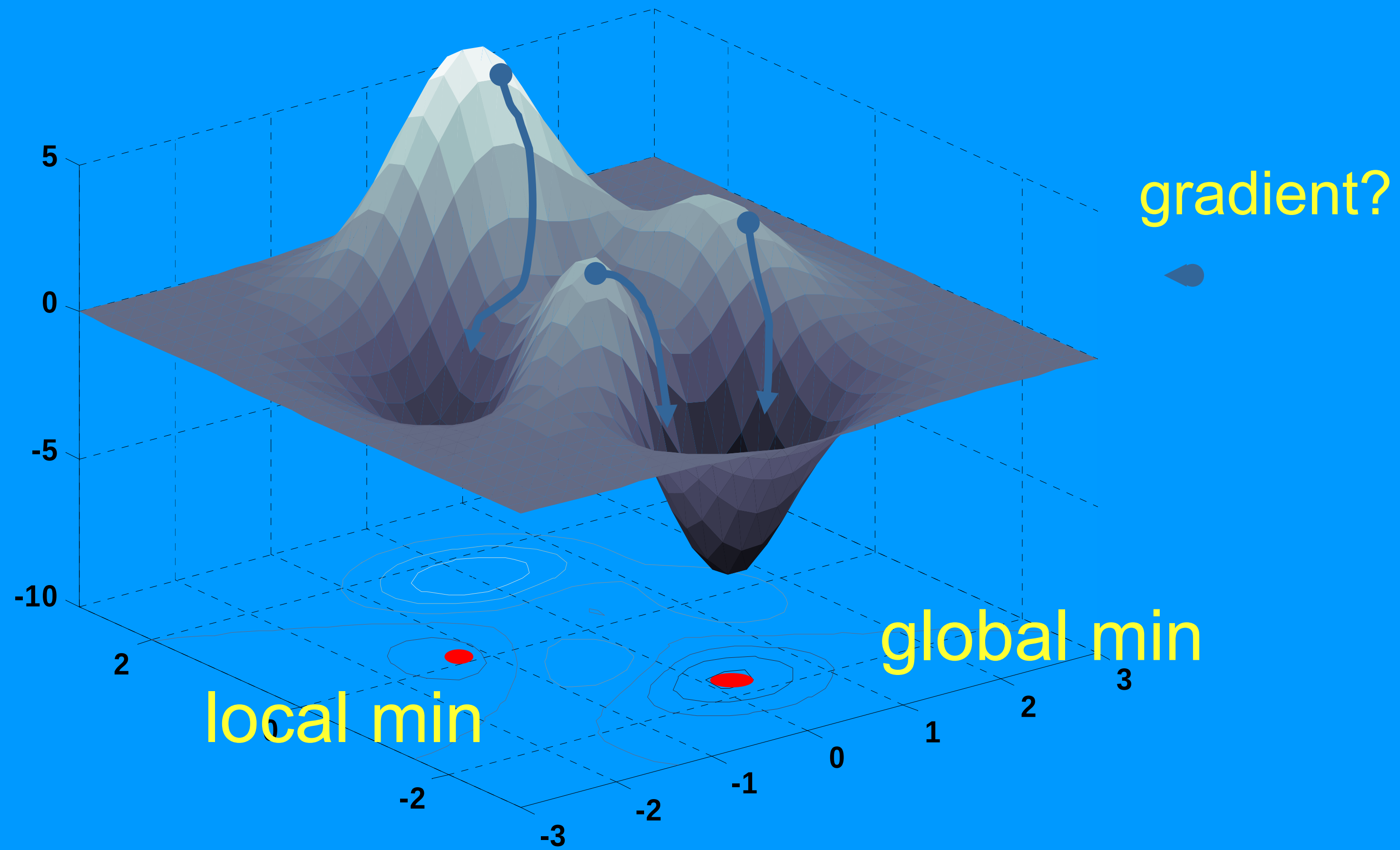
$$dw/dt \propto -\partial J/ \partial w$$

$$w \leftarrow w - \eta \, \partial J/ \partial w$$



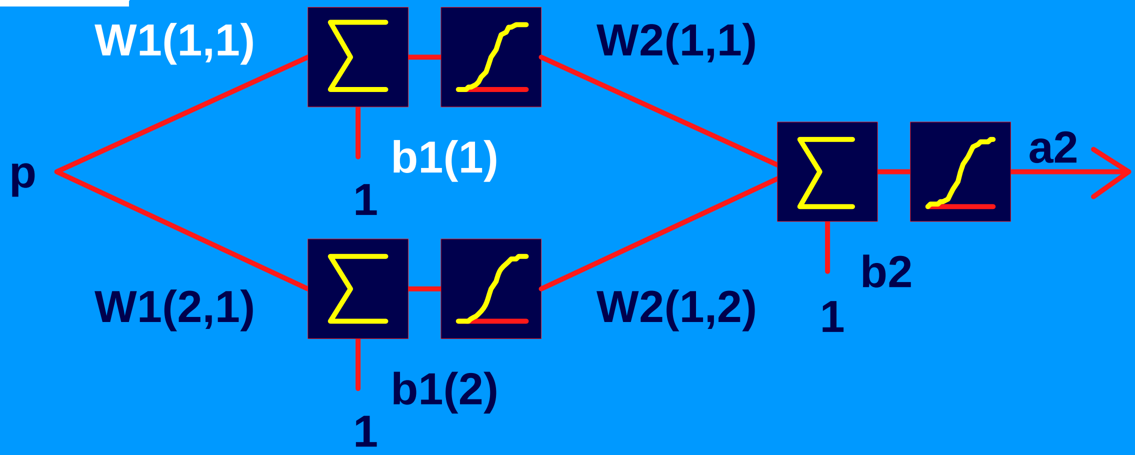

∂J/ ∂w = 0
local
minimum
∂J/ ∂w = 0
cost, J
global
minimum
∂J/ ∂w = 0
weight, w

Multi-Modal Cost Surface

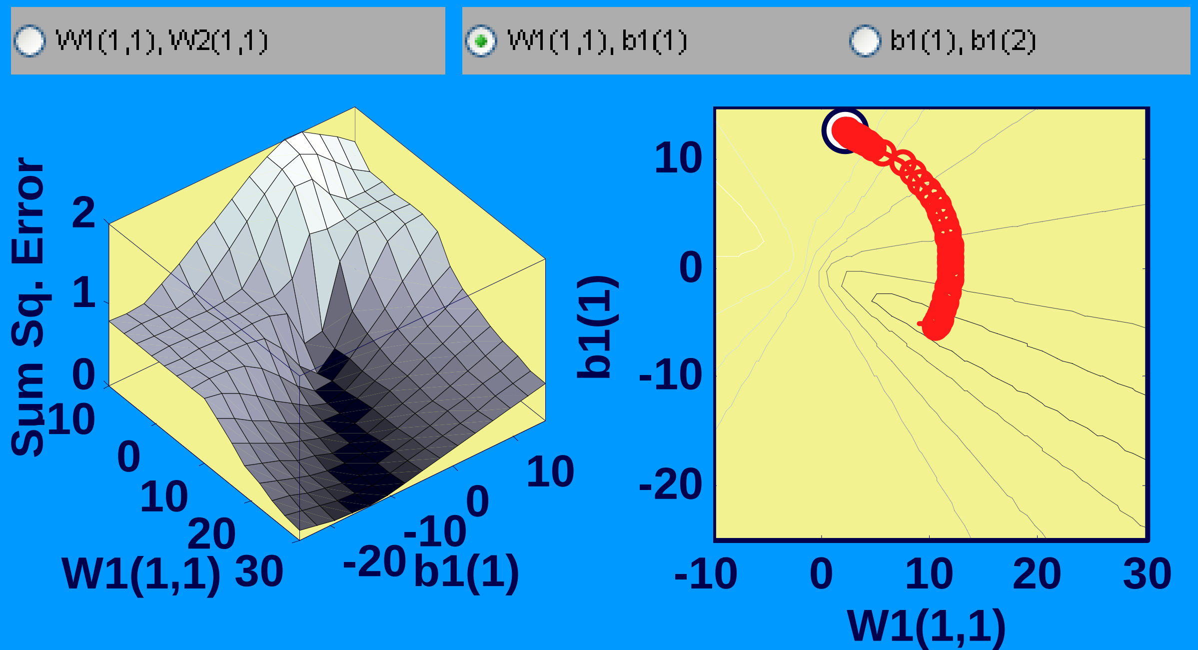


Heading Downhill

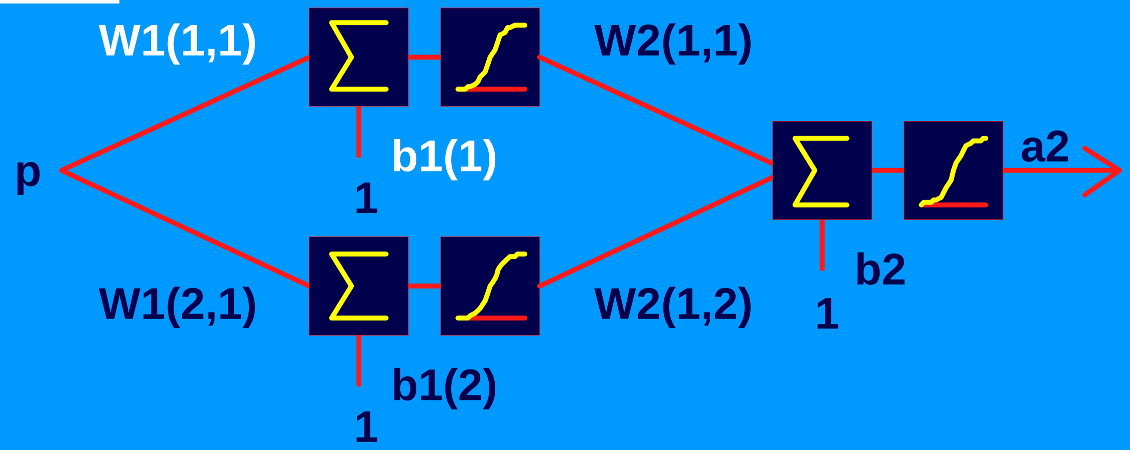
- assume
  - minimization (e.g. SSE)
  - analytically intractable
- step parameters downhill
- $w_{new} = w_{old}$ + step in right direction
- backpropagation (of error)
  - slow but efficient
- conjugate gradients, Levenburg/Marquardt
  - for preference

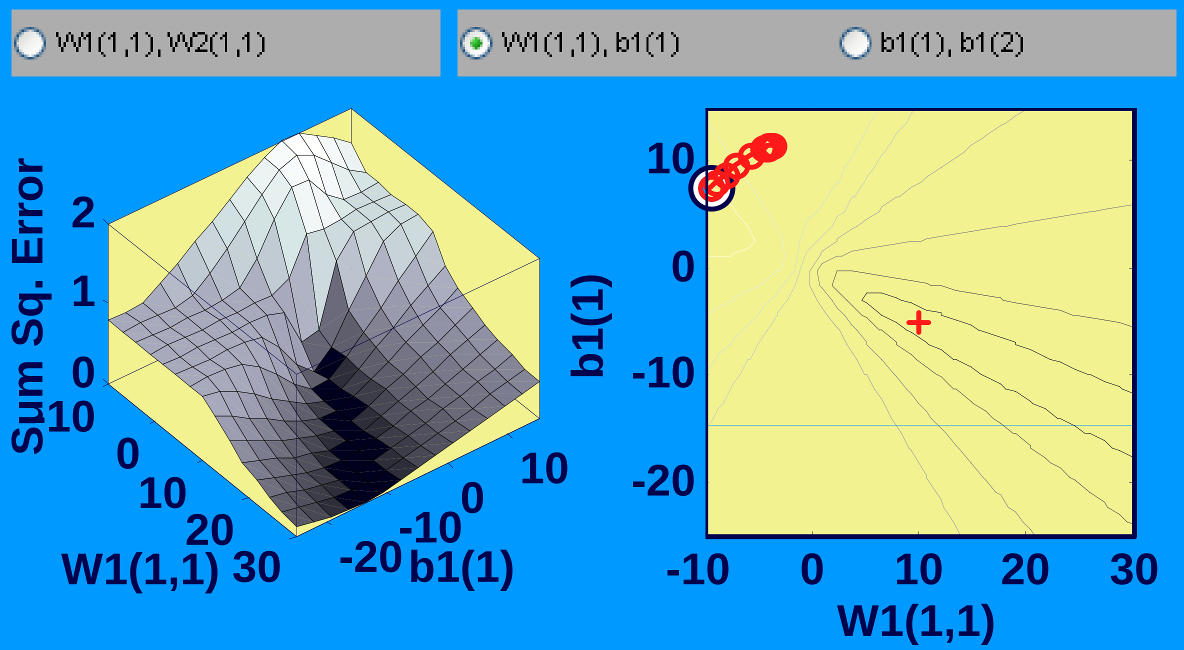


W1(1,1)
W2(1,1)
p
b1(1)
1
a2
b2
1
W1(2,1)
W2(1,2)
b1(2)
1



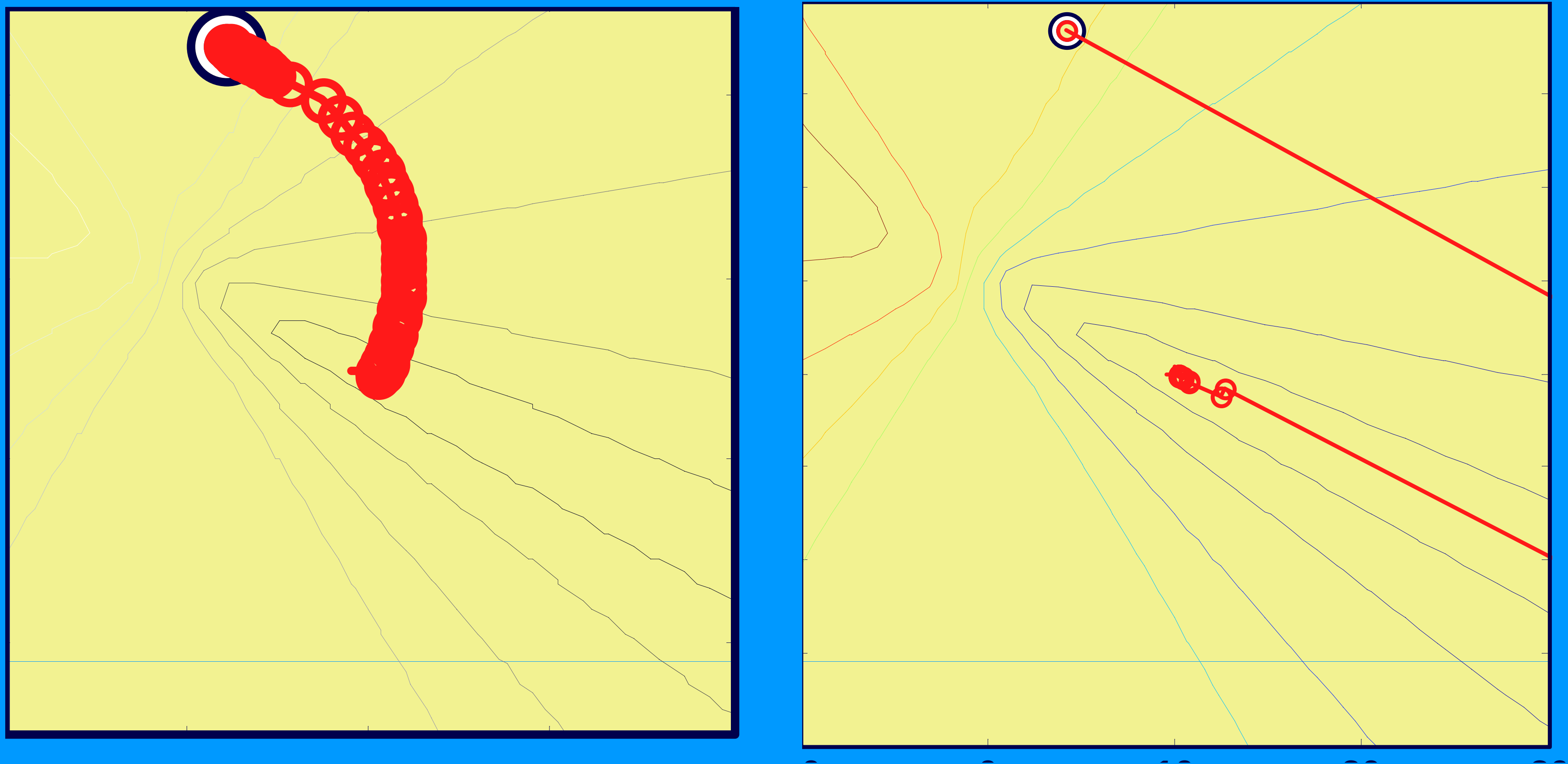
W1(1,1), W2(1,1)
W1(1,1), b1(1)
b1(1), b1(2)
Sum Sq. Error
2
1
0
-10
0
10
20
W1(1,1) 30
10
0
-10
-20
b1(1)
b1(1)
10
0
-10
-20
-10
0
10
20
30
W1(1,1)


W1(1,1)
W2(1,1)
p
b1(1)
1
a2
W1(2,1)
W2(1,2)
b2
1
b1(2)
1


W1(1,1), W2(1,1)
W1(1,1), b1(1)
b1(1), b1(2)
Sum Sq. Error
2
1
0
-10
0
10
20
30
W1(1,1)
-20
-10
0
10
b1(1)
b1(1)
10
0
-10
-20
-10
0
10
20
30
W1(1,1)

backprop

conjugate gradients



Implications

- correct structure can get “wrong” answer
  - dependency on initial conditions
  - might be good enough
- train / test (cross-validation) required
  - is poor behaviour due to

    network structure?

    ICs?

additional dimension in development

The University Of Sheffield.

RBF NN Warning!

- RBF NNs claimed to have unique solution

BUT

- who picks the pre-processing layer?
  - direct optimisation of centres and widths
  - some other method
- L.I.P. models have “non-linear parameters” to select → multi-modal cost ≡ MLP

Are multiple minima a problem?

- pros seem to outweigh cons
- good solutions often arrived at quickly
- all previous issues apply
  - sample density & distribution
  - lack of prior knowledge

How to Use

- to “generalize” a GLM
    - linear regression – curve-fitting

      linear output + SSE
    - logistic regression – classification

      logistic output + cross-entropy (deviance)

      extend to multinomial, ordinal

        e.g. softmax outptut + cross entropy
    - Poisson regression – count data

What is Learned?

- the right thing
  - in a maximum likelihood sense

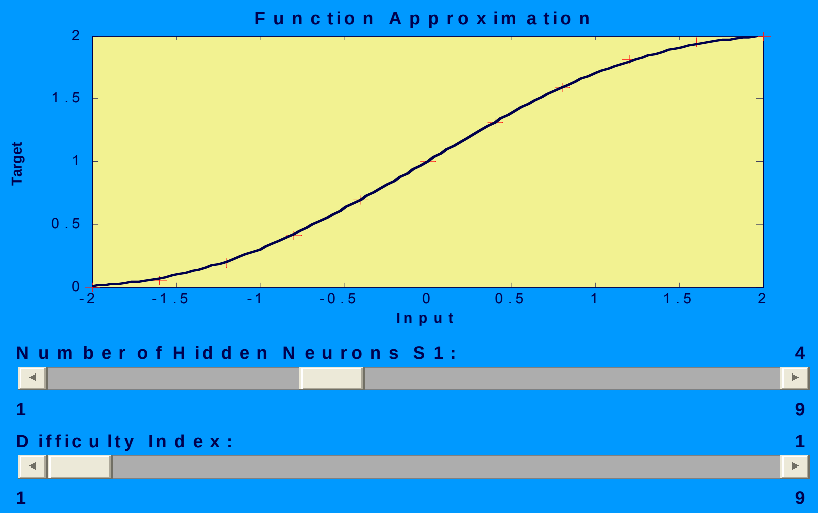
theoretical

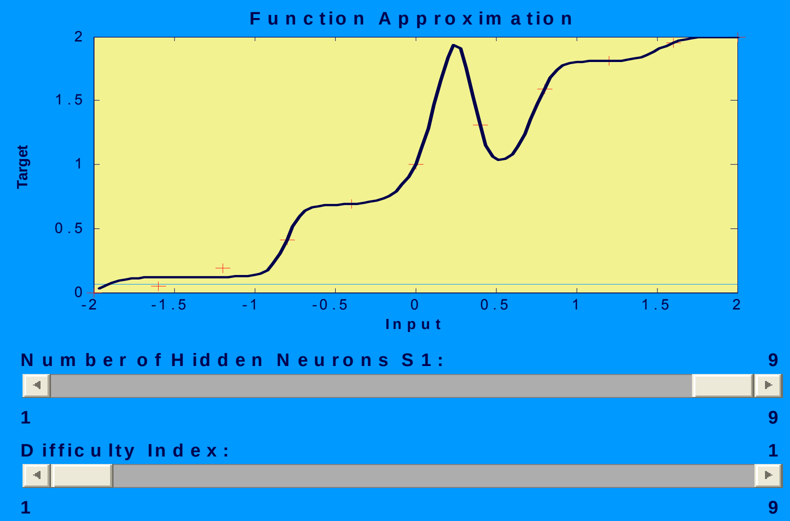
- conditional mean of target data, $E(z|x)$
  - implies probability of class membership for classification $P(C_i|x)$

estimated

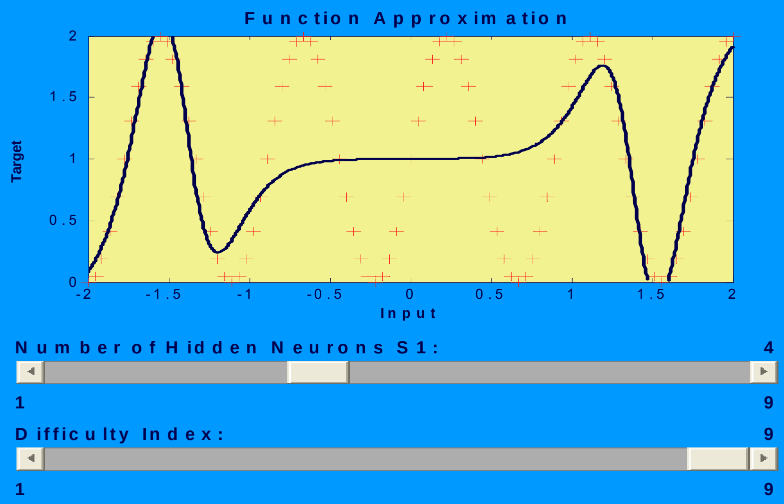
- if good estimate then $y \rightarrow E(z|x)$

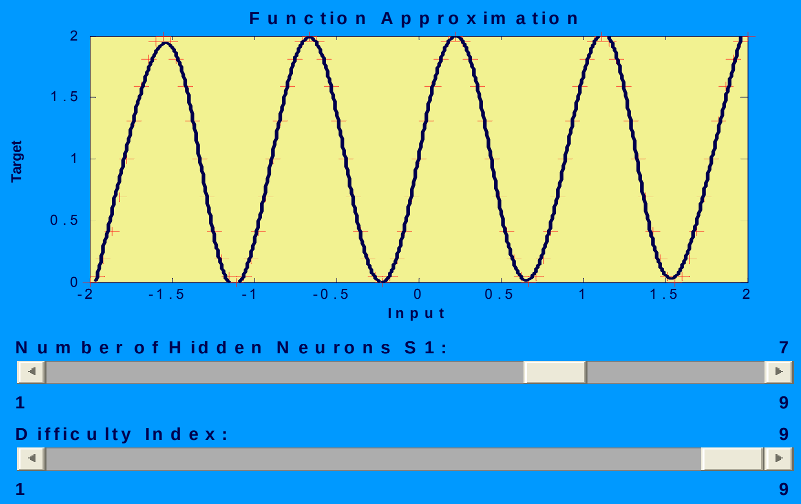
Simple 1-D Function



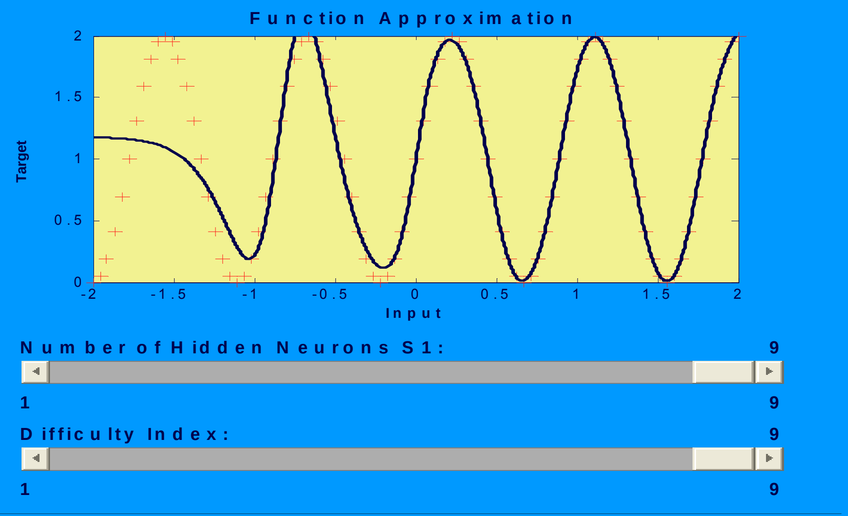


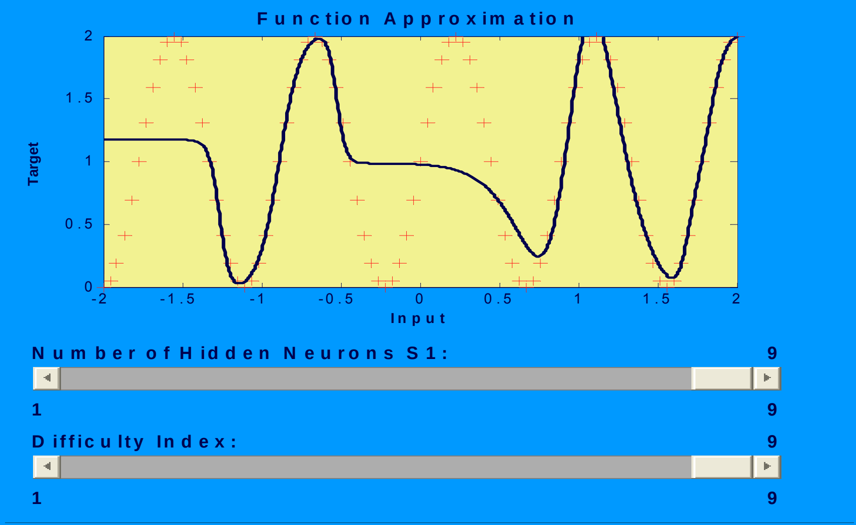
More Complex 1-D Example



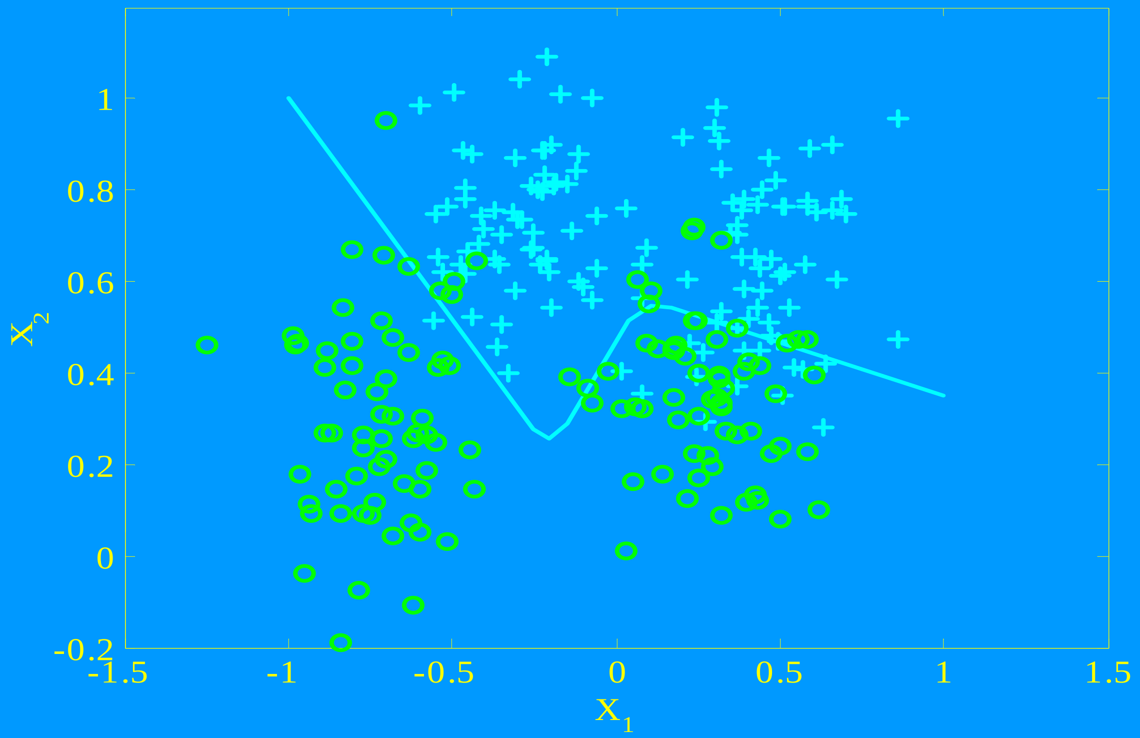


Local Solutions



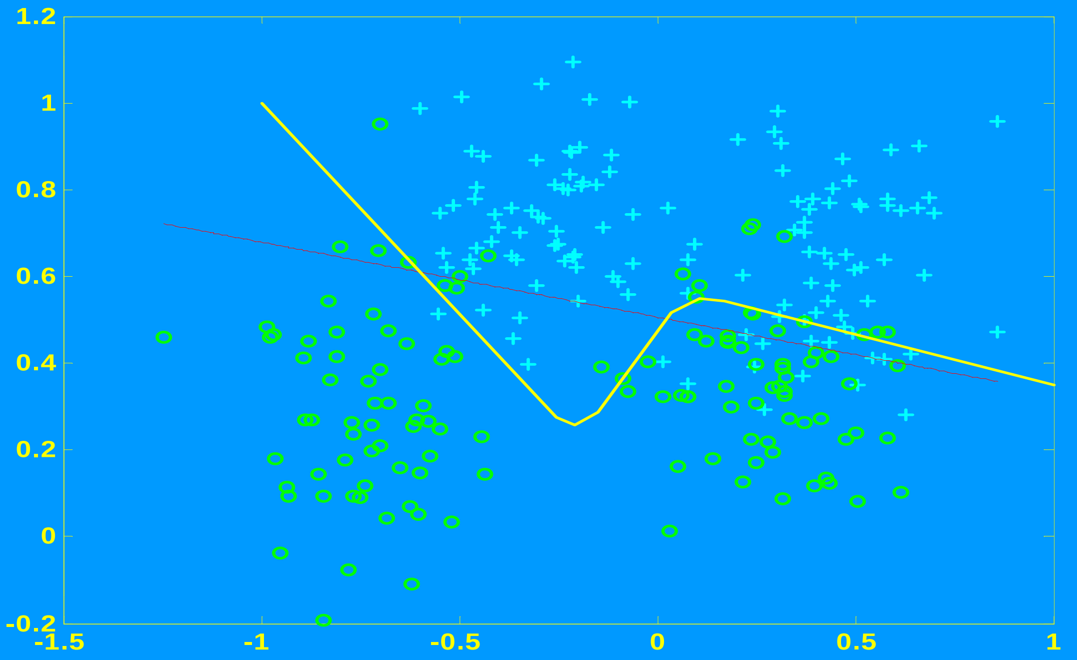


A Dichotomy



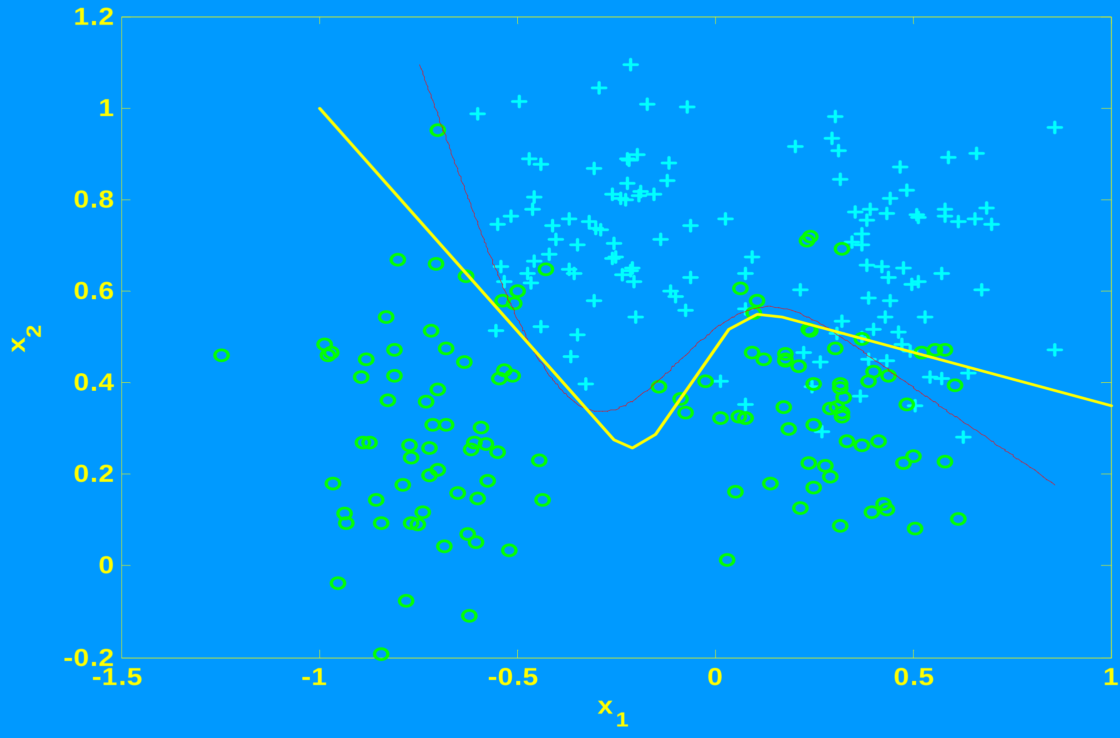
data courtesy B Ripley

Linear Decision Boundary



induced by $P(C_1|x) = P(C_2|x)$

Non-Linear Decision Boundary



Over-fitted Decision Boundary