Non-linear Modelling by Adaptive Pre-processing

Rob Harrison
Automatic Control & Systems Engineering
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
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)

zero error?

rough components
Underfitting (sample data)

over-smooth components

EPSRC Winter School January 2008
Goldilocks

v. small error

“just right” components
Restricting “Flexibility”

- use data to tell the estimator how to behave
- regularization/penalization
  - penalize “roughness”
  - e.g. $\text{SSE} + \rho Q$
- $Q = \sum w_{ij} \rightarrow \hat{w} = (\Phi^T\Phi + \rho I)^{-1} \Phi^T z$
- use potentially complex structure
  - data constrains where it can
  - $Q$ constrains elsewhere
Hold-out Method

keep back P% for testing
wasteful
sample dependent

Training RMSE 0.23
Testing RMSE 0.38
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. $\rho$, number, width …
EPSRC Winter School January 2008
generalization error
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

output, y

adaptive layer

target, z

error, e
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

\[ y = \theta \left( w^T y + b \right) \]
\[ v_j = \sigma \left( w^T x + b_j \right) \]
\[ y = \theta \left( \sum_i w_i \sigma \left( \sum_j w_j x_j + b_j \right) + b \right) \]
A Sigmoidal Unit

\[ y_j = f\left( a_j \right) = f\left( b_j + \sum_i w_{ji} x_i \right) \]
Combinations of Sigmoids

EPSRC Winter School January 2008
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

\[
J_\infty = \frac{1}{2} \sum_{i=1}^{n} \int_{x \in \mathbb{R}^d} \left( E[z_i | x] - \hat{f}_i(x; \mathcal{W}) \right)^2 p(x) \, dx \\
+ \frac{1}{2} \sum_{i=1}^{n} \int_{x \in \mathbb{R}^d} \left( z_i - E[z_i | x] \right)^2 p(x) \, dx
\]

MLP

\[
\hat{f}_i(x; \mathcal{W}^*) \square E[z_i | x]
\]
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

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

\[ \delta_j(t) = \theta'(net_j(t)) e_j(t) \quad m = M \]  
output layer

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

- parameter → cost “malign”
  - optimization difficult
  - many solutions possible
  - effect of hidden weights in output non-linear
Rolling Ball

unique minimum
\[ \frac{\partial J}{\partial w} = 0 \]
Gradient Descent

\[
\frac{dw}{dt} \propto -\frac{\partial J}{\partial w}
\]

\[w \leftarrow w - \eta \frac{\partial J}{\partial w}\]

unique minimum
\[\frac{\partial J}{\partial w} = 0\]
\[ \frac{\partial J}{\partial w} = 0 \]

**local minimum**
\[ \frac{\partial J}{\partial w} = 0 \]

**global minimum**
\[ \frac{\partial J}{\partial w} = 0 \]
Multi-Modal Cost Surface

Gradient?

Local min

Global min
Heading Downhill

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

Number of Hidden Neurons S1: 4
Difficulty Index: 1

Number of Hidden Neurons S1: 9
Difficulty Index: 1
More Complex 1-D Example

Function Approximation

Number of Hidden Neurons S1:

4

1

Difficulty Index:

9

Number of Hidden Neurons S1:

7

1

Difficulty Index:

9

EPSRC Winter School January 2008
Local Solutions

EPSRC Winter School January 2008
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