Making the most of finite training sets

- Background
- Improving balance between bias and variance
- Outlier detection in feed-forward nets
- Example
- Bimodal Distribution Removal algorithm
- Comparisons and other approaches
  - Absolute Criterion
  - Least Median Squares
  - Least Trimmed Squares
- Heuristic Pattern Reduction
- Error Sign Testing

Background

- Parametric estimators
  - efficient - time, data reqd.
  - must know model
- Non-parametric
  - model free
  - potentially infinite data reqd.
- Feed-forward network
  - non-parametric
  - guaranteed to outperform param. estimators as training set $\rightarrow \infty$
- Bias and Variance dilemma
  - high bias - predispose to incorrect solution
  - high variance - sensitivity to data
  - low bias low variance - large amount of data
  - increase bias to reduce amount of data reqd.
  - balance bias and variance

Bias-variance decomposition

- Regression prob.: construct function based on training set
  - $f(\mathbf{x}; D)$ f depends on training data D
- Mean squared error of f as estimator of regression is
  - $E_D\left[\left(f(\mathbf{x}; D) - E[y \mid x]\right)^2\right]$
- Bias-variance decomposition of estimator is
  - $\left(E_D[f(\mathbf{x}; D)] - E[y \mid x]\right)^2$ "bias"

    $+E_D\left[\left(f(\mathbf{x}; D) - E_D[f(\mathbf{x}; D)]\right)^2\right]$ "variance"

  - If the expected value of the function is different on average from the regression, then it is biased.
  - If the function has large differences from the expected value it has high variance.
- Estimated bias (over 50 test patterns) $\text{Bias}(x) \approx \frac{1}{50} \sum_{i=100}^{150} \left|\overline{f}(x_i, w) - y_i\right|^2$
- Estimated variance $\text{Variance}(x) \approx \frac{1}{50} \sum_{i=100}^{150} \frac{1}{50} \sum_{j=1}^{50} \left|f(x_j, w; D_i) - y_i\right|^2$

Improve balance of bias & variance

- Pruning network
  - class of functions network can recognise is reduced
  - increases bias
- Growing network
  - small initial network has high bias
- Extra terms in performance function
  - smoothing decreases variance at the cost of increasing bias
- Cross validation
  - halt training when test set performance degrades
  - restricts network learning of training set
- Outlier removal
  - reduces noise and hence the variance of training set

Outlier detection in feed-forward nets

- Normal back-propagation
  - minimise mean square error
- Absolute Criterion
  - minimise lower absolute value of error
  - outliers never used ⇒ better generalisation
  - must know how many outliers exist
  - prone to oscillation
- Least Median Squares
  - minimise median of error
  - slow to converge
- Least Trimmed Squares
  - minimise only lowest mean square errors
  - outliers never used ⇒ better generalisation
  - must know how many outliers exist
  - does not perform well on real noisy data
- Bimodal Distribution Removal / Heuristic Pattern Reduction

Example

- 150 patterns - class results, undergraduate Computer Science, COMP1111 at University of NSW
- raw data = results from:
  - laboratory exercises
  - assignments
  - mid-term quiz
  - 40% of mark
  - final aggregate mark
- goal: predict final mark based on partial marks
- educational imperative - reliable prediction of mark based on their current performance
- expect to invalidate the prediction -> students will improve their preformance

Bimodal error distribution

- Error distribution before training

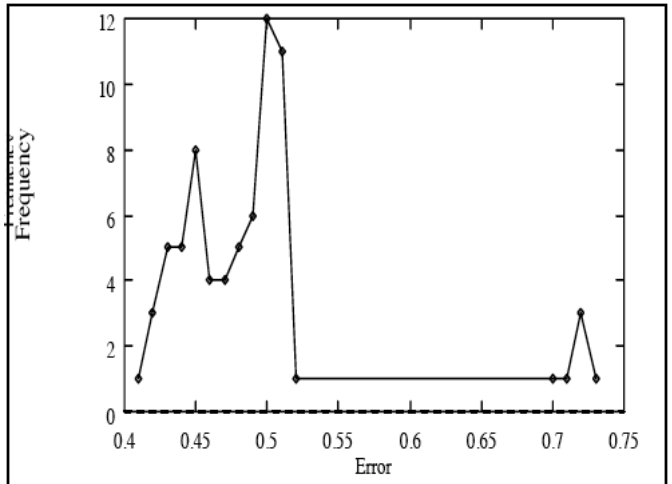

Figure 1. : Error distribution at epoch 0

- Error distribution after some training

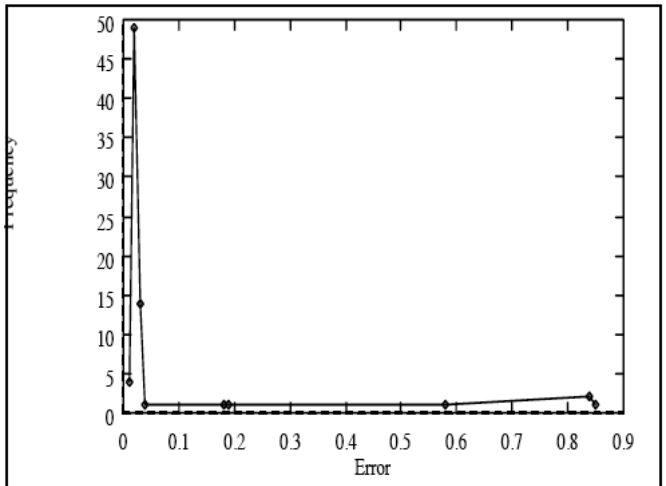

Figure 2. : Error distribution at epoch 500

Bimodal Distribution Removal algorithm:

- Begin training with the whole training set.
- Wait until the normalised variance of errors over the training set υ_{ts}, is below 0.1.
- Calculate the mean error $\overline{\delta_{ts}}$, over the training set.
- Take from the training set those patterns error greater than $\overline{\delta_{ts}}$.
- Calculate the mean $\overline{\delta_{ss}}$, and standard deviation σ_{ss} of this skewed subset.
- Permanently remove all patterns from the training set with error $\geq \overline{\delta_{ss}} + \alpha\sigma_{ss}$ where $0 \leq \alpha \leq 1$.
- Repeat steps 2-6 every 50 epochs, until normalised variance of errors over the training set $\upsilon_{ts} \leq 0.01$.
- Halt training.

Comparisons

- Normal back-prop

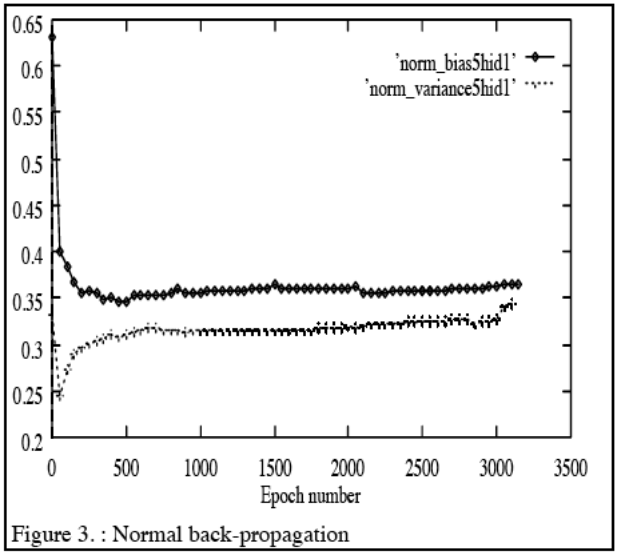

Figure 3. : Normal back-propagation

- Bimodal distribution removal

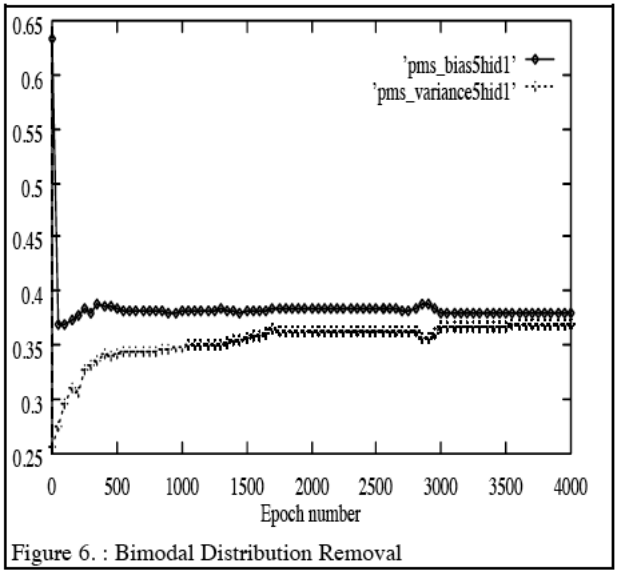

Figure 6. : Bimodal Distribution Removal

Comparisons

- Least Trimmed Squares

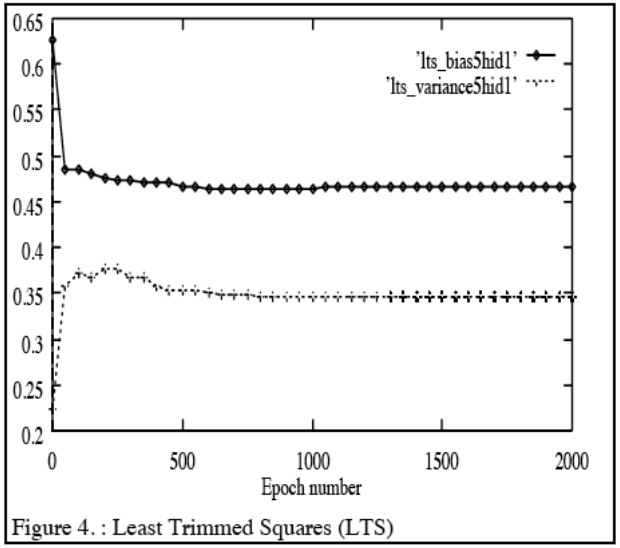

Figure 4. : Least Trimmed Squares (LTS)

- Heuristic Reduction

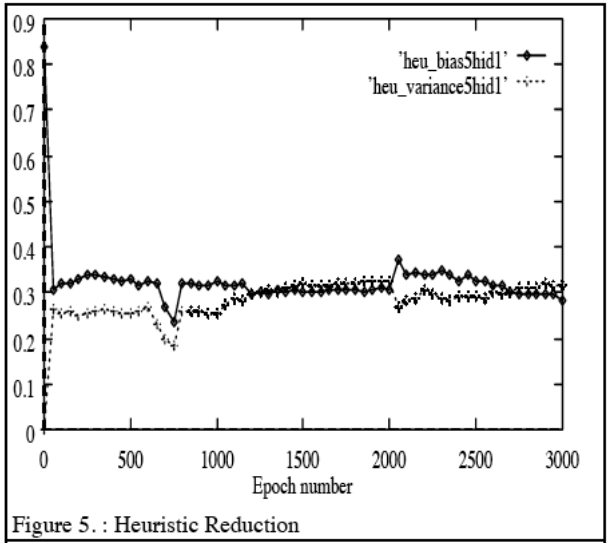

Figure 5. : Heuristic Reduction

Heuristic pattern reduction

- Network architecture
  - 14 input 5 hidden 4 output (1 per grade: F, P, CR, D)
- Patterns
  - Test set - 50 patterns (random) training set - 100 patterns
- Adjacency
  - 1D - contribution to total sum of squares
- Assumption
  - homogenous distribution
  - clusters of training patterns
  - pairs of patterns (training set reduced to half size)
- Testing
  - 15 runs of each configuration; different initial weights

Sample set

	run no.				
number of patterns	sec6	sec7	sec8	sec9	sec10
100	37.98	28.72	34.49	35.55	40.82
80	30.42	46.02	34.86	36.70	43.11
67	44.05	49.01	49.57	32.54	36.98
50	42.26	52.69	43.83	46.73	42.59
33	37.22	48.36	36.55	35.13	43.89
25	36.71	41.12	55.53	40.83	29.80

Summary of results

Number of Pats	Tss value	
100	28.72	← full training set
80	27.38	
67	30.48	
50	24.78	← best value
33	32.90	
25	34.69	

Conclusions

- Heuristic pattern reduction conclusion:
  - simple general heuristic method
  - reduce size of training pattern set
  - improvement of performance on validation set
    - improvement likely due to simplification of error surface in pattern space / removal of some outliers
  - no significant features of original pattern set lost
  - fewer training patterns
    - speed up training
- Bimodal Distribution Removal conclusion:
  - As good as other methods on ‘real’ (noisy) data