A hybrid projection based / radial basis functions

Shimon Cohen **Nathan Intrator**

Tel-Aviv University and Brown University

www.physics.brown.edu/users/faculty/intrator

Motivation

Previous talk:

- Use over-complex architecture (little bias)
- Address the resulting variance by injecting independence and averaging

Motivation

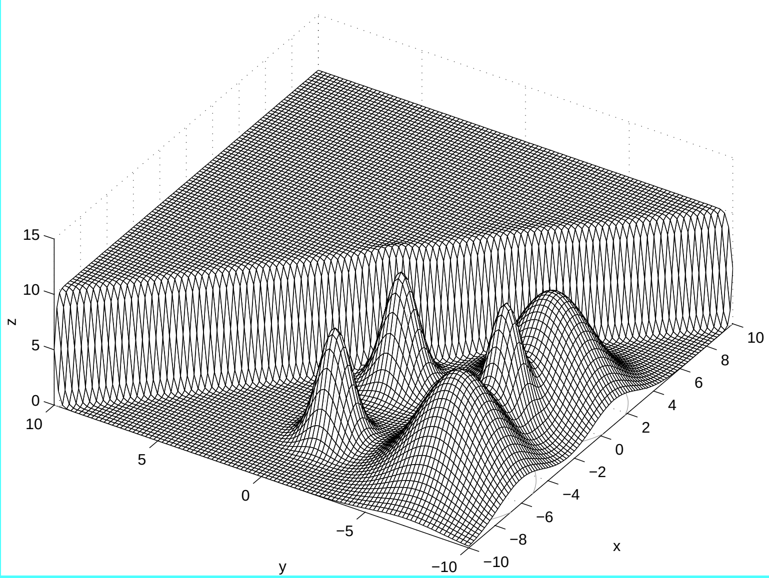
Previous talk:

- Use over-complex architecture (little bias)
- Address the resulting variance by injecting independence and averaging

This talk:

- Use very compact architecture (small variance)
- Attempt to fit the data as best as possible (low bias)

Hybrid Architecture: Fitting the data better

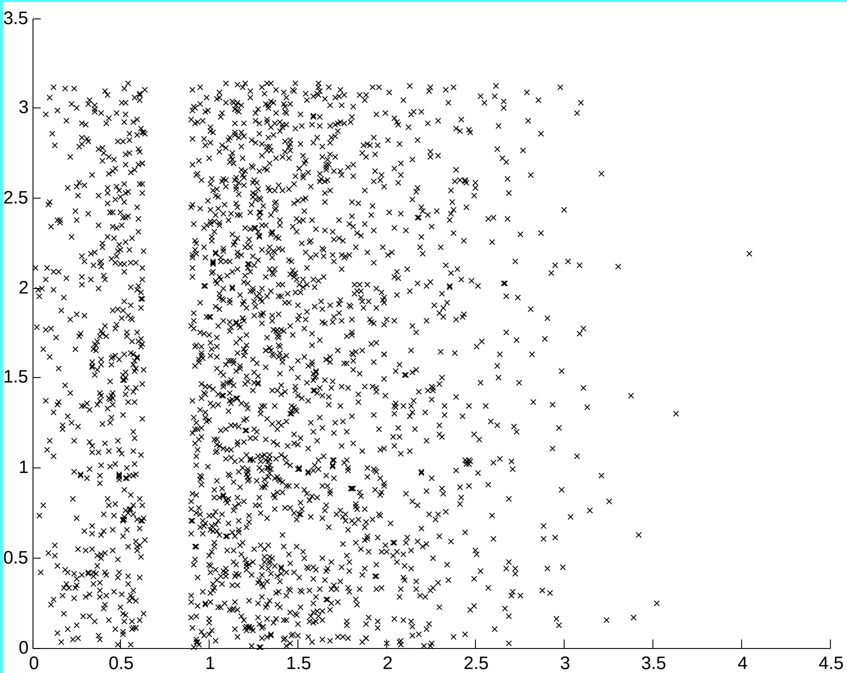


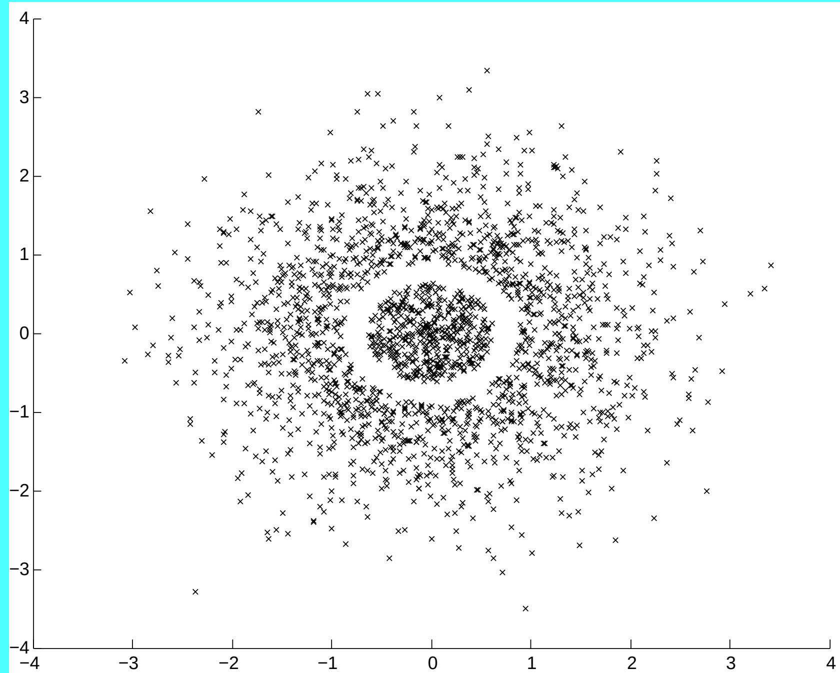
$z = f(x, y)$ is composed of five clusters and a sigmoidal surface.

- Data complexity: not homogenous across regions
- Linear, Sigmoidal and Gaussian regions

Requires a **divide and conquer** approach with different complexity architecture.

Type of hidden units





MLP and RBF are complimentary units

"A function can be decomposed into mutually exclusive radial and projection based parts"

(Donoho and Johnstone, 89)

Background

Previous work on flexible estimators that include Ridge and RBF functions:

- Generalized additive models (Hastie & Tibshirani, 90)

- Higher-Order Networks (Lee et al., 86)

$$a_j = g(\sum_i (w_{ji} \cdot x_i) + \sum_k \sum_l w_{ikl} x_k x_l).$$

- Adding a squared version of the inputs (old statistical idea) SMLP (Flake, 98):

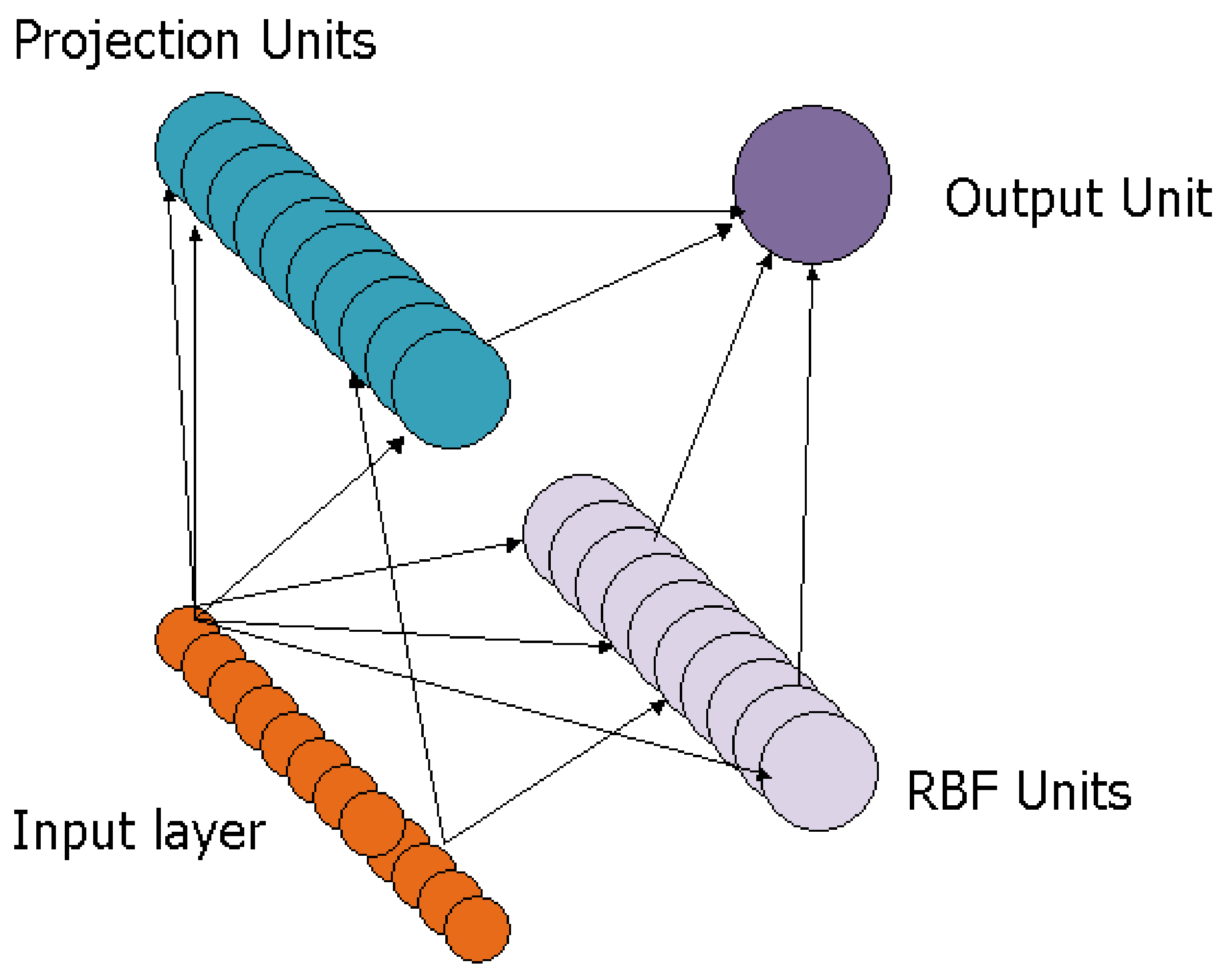
$$a_j = g(\sum_i (w_{ji} \cdot x_i) + \sum_k \sum_l w_k x_k^2).$$

Classical approach

First find radial part and then projection part on the residual error

Problem: Difficult to recover from residuals (caused by bad approximators)

Hybrid RBF/BP Architecture (PRBFN)



Hidden unit outputs

Projection units:

$$a_j = \sigma(\sum_i (w_{ji} \cdot x_i)).$$

Radial basis unit:

$$\phi(x, w_i) = \mathrm{exp}^{-(x-w_i)^2/(2r_i^2)}.$$

Challenges: Automatic Architecture Selection

- Determine network size and unit type

- Computational efficiency (no retraining)

Network construction & training procedure

- Decompose the input space into homogenous regions
- Choose the appropriate unit for each specific region in input space
  - Includes determination of initial weights
- Determine network size (prune)
- Train the full network

Network construction & training procedure

- Decompose the input space into homogenous regions
- Choose the appropriate unit for each specific region in input space
  - Includes determination of initial weights
- Determine network size (prune)
- Train the full network

Input space division

- A CART like algorithm:
- Recursively divide current input space into two sub regions
- Choose two anchor points:

$$
\begin{aligned}
x_1 &= \arg\max_x f(x) \quad C_1 = \{x : d(x, x_1) < d(x, x_2)\} \\
x_2 &= \arg\min_x f(x) \quad C_2 = \{x : d(x, x2) < d(x, x1)\}
\end{aligned}
$$

Input space division (continued)

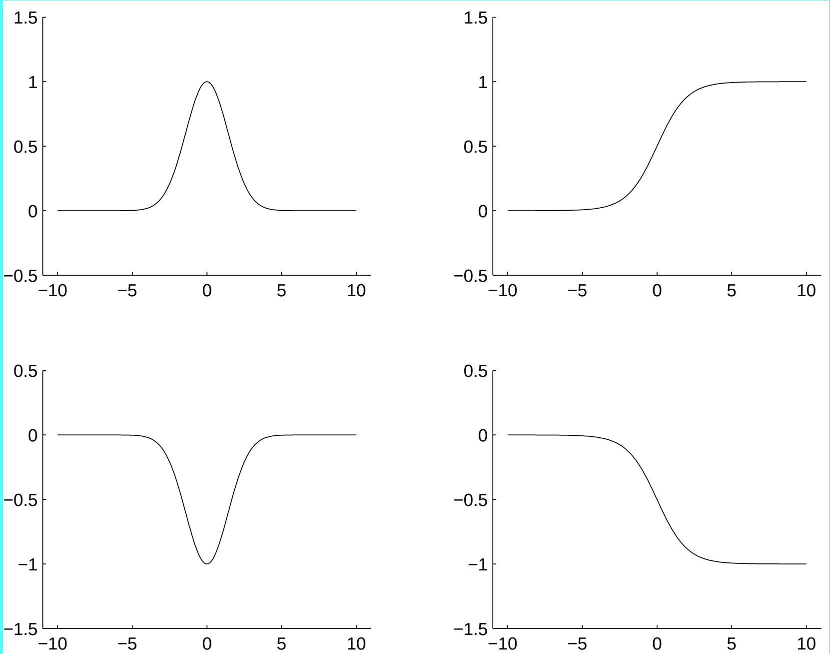
- Objective function:

$$SSR(C_0) = \sum_{y_i \in C_0} (y_i - \bar{yo})^2,$$

- Maximum reduction in:

$$\Delta SSR(C_0) = SSR(C_0) - (SSR(C_1) + SSR(C_2)).$$

Unit type Selection



Left: RBF, right: ridge (positive and negative)

Hidden unit weights

- RBF unit: set center at the maximum point of the subspace.

- Projection unit: set the weight vector to be normalized and maximal at the maximum point of the subspace.

Unit type selection via the Evidence

- The Bayes Factors are defined as:

$$\frac{p(M_1|D)}{p(M_2|D)} = \frac{p(D|M_1)p(M_1)}{p(D|M_2)p(M_2)}.$$

- Integrating the unknown weights:

$$\begin{aligned} p(D|M) &= \int_W p(D,W|M)dW \\ &= \int_W p(D|W,M)p(W|M)dW. \end{aligned}$$

- The integration can be performed by using Laplace integral, (Taylor approximation to the second order).

$$p(D|M) \cong (2\pi)^d |H|^{-1|2} p(D|W_{m_0}, M) p(W_{m_0}|M)$$

Unit type selection (objective)

Choose the model (RBF or MLP) which maximizes the likelihood. Or:

- Assume: Gaussian noise on the targets $N(0, \alpha^2)$, and Gaussian prior on the weights: $N(0, \beta^2)$.

- Let $y_i = w\phi(x_i) + w_o$, where ϕ is either an RBF or MLP. consider:

$$L = \frac{1}{(2\pi)^{N/2}\alpha^N} exp(\frac{-\sum_{i=1}^{N}(y_i - t_i)^2}{2\alpha^2}) \frac{1}{(2\pi)^{1/2}\beta} exp(-\frac{W^T W}{2\beta^2}).$$

- Consider the log of L (ignoring constants)

$$LL = N \log(\alpha) + \frac{\sum_{i=1}^{N}(y_i - t_i)^2}{2\alpha^2} + \log(\beta) + \frac{W^T W}{2\beta^2}.$$

Unit type selection overview

For

$$LL = N\log(\alpha) + \frac{\sum_{i=1}^{N}(y_i(w,w_0)-t_i)^2}{2\alpha^2} + \log(\beta) + \frac{W^TW}{2\beta^2},$$

set the gradient of LL to zero with respect to α, β, w, w_0 and find optimal values.

Given optimal values, select the model with highest MAP.

Unit type selection (details)

- set $\nabla_{\alpha,\beta} LL = 0$, to obtain

$$\alpha^2 = \frac{\sum_{i=1}^{N}(y_i - t_i)^2}{N}.$$

$$\beta^2 = W^T W.$$

- MLE minimizes the error only, without penalizing on model complexity (small weights)

Unit type selection (continued)

- Differentiating LL with respect to w_0 gives:

$$w_0 = \frac{1}{N}(\sum_{i=1}^{N}(yi - w\sum_{i=1}^{N}\phi_i).$$

- Differentiating LL with respect to w gives:

$$w = \frac{\beta^2 \sum_{i=1}^{N} t_i\phi_i - \frac{\beta^2}{N}\sum_{i=1}^{N} t_i \sum_{i=1}^{N}\phi_i}{\beta^2 \sum_{i=1}^{N}\phi_i^2 - \frac{\beta^2}{N}\sum_{i=1}^{N}\phi_i \sum_{i=1}^{N}\phi_i + \alpha^2}.$$

Unit type selection (continued)

The Hessian of the negative log-likelihood is given by:

$$\mathbf{H} = \begin{pmatrix} \frac{\sum_{i=1}^{N} \phi_i^2}{\alpha^2} + \frac{1}{\beta^2} & \frac{\sum_{i=1}^{N} \phi_i}{\alpha^2} \\ \frac{\sum_{i=1}^{N} \phi_i}{\alpha^2} & \frac{N}{\alpha^2} \end{pmatrix}.$$

Using

$$LL = N \log(\alpha) + \frac{\sum_{i=1}^{N} (y_i - t_i)^2}{2\alpha^2} + \log(\beta) + \frac{w^2}{2\beta^2},$$

and the Gaussian approximation:

$$p(D|M) \cong (2\pi)^d |H|^{-1|2} p(D|W_{m_0}, M) p(W_{m_0}|M),$$

the log of the evidence becomes:

$$LL = -N \log(\alpha) - \log(\beta) - \frac{1}{2} \log(|H|).$$

Unit selection algorithm

- Initialize α and β.
- Loop: compute w, w_o and α, β using the previous derivation
- Stop when α converges ($\Delta\alpha$) is small.
- Based on α, β and H, select the unit with highest MAP:

$$LL = -N \log(\alpha) - \log(\beta) - \frac{1}{2} log(|H|).$$

$H \simeq Ni$ and $\frac{1}{2} log(|H|) = O(N^{-.5})$

Network construction & training procedure

- Decompose the input space into homogenous regions
- Choose the appropriate unit for each specific region in input space
  - Includes determination of initial weights
- Determine network size (prune)
- Train the full network

Pruning using a Gaussian error model

- Assume that the target function values are corrupted by Gaussian noise with zero mean and equal variance σ^2.

- Assume that the patterns in the training set are independent, the likelihood of the data under the model is

$$L = \frac{1}{(2\pi)^{\frac{N}{2}}\sigma^N} \exp(-\frac{\sum_{n=1}^N (y_n - t_n)^2}{2\sigma^2}).$$

- For maximization, consider the log value of L:

$$LL = -\frac{N}{2}\log(2\pi) - N\log(\sigma) - \frac{\sum_{n=1}^N (y_n - t_n)^2}{2\sigma^2}.$$

- The maximum likelihood with respect to σ is:

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(y_n - t_n)^2.$$

Likelihood Ratio Test (for pruning)

- The LRT can be used to select between two nested models.

- Given two models $M1 \subset M2$ the

$$-2\log\left(\frac{p(D, W_{m_0}|M1)}{p(D, W_{m_0}|M2)}\right) \sim \chi^2(d_2 - d_1).$$

- Uses $P - Values$ to reject the null hypothesis, that is, the simple model is equivalent to the complicated one.

- Applicable only for pruning.

Bayesian Information Criteria (BIC) approximation

- The BIC approximation can be derived, by using Gaussian distribution to the a-priori parameters density to arrive at:

$$BIC \equiv \log(p(D|M)) \; = \; \log(p(D, W_{m_0}|M)) - \frac{d}{2}\log(N),$$

where $\log(p(D, W_{m_0}|M))$ is the MLE, N, the number of points, and d is the number of parameters.
(Schwartz 78, Kass & Raftery, 95)

Pruning algorithm summary

- Find $\hat{\sigma}_1$ and $\hat{\sigma}_2$ for each model using the MLE. The LRT becomes:

$$\chi^2(d2 - d1) \simeq 2N \log(\hat{\sigma}_1^2) - 2N \log(\hat{\sigma}_2^2).$$

- Apply P values to reject the null (small model is better)

- Similarly, BIC becomes:

$$BIC_i = -N \log(\hat{\sigma}_i) - \frac{d_i}{2} \log(N), \quad i = 1, 2.$$

- Choose the larger BIC.

Final global training

- Divide input space and assign units to each sub-region.
- Select type of hidden unit for each sub-region (and initial values).
- Stop when error goal, or maximum number of units, is achieved.
- Prune un-necessary weights.
- Full Global optimization.

The final global optimization can remove overfitting caused by data driven subspace division.

Application: Function approximation (Clustering)

Clusterization for Function Approximation (CFA) Data was taken from (Gonzalez et al. 2002).

- Three data sets.
- CFA is used at the first stage for RBFN.
- Study the normalized root mean square error (NRMSE):

$$\text{NRMSE} = \sqrt{\sum_{i=1}^{n} [f(x^i) - t^i]^2 / \sum_{i=1}^{n} [f(x^i) - \bar{t}]^2},$$

CFA Application (continued)

- The first target to approximate is:

$$f_1(x) = \frac{sin(2\pi x)}{exp(x)}, x \in [0, 10].$$

- Four prototypes and 1000 samples of f_1 generated by evaluating inputs taken uniformly from the interval $[0, 10]$.

- The second function, also taken from CFA, to consider is:

$$f_2(x) = 0.2 + 0.8(x + 0.7\sin(2\pi x)), x \in [0, 1]$$

  from 21 equidistant input-output training examples belonging to the interval $[0, 1]$.

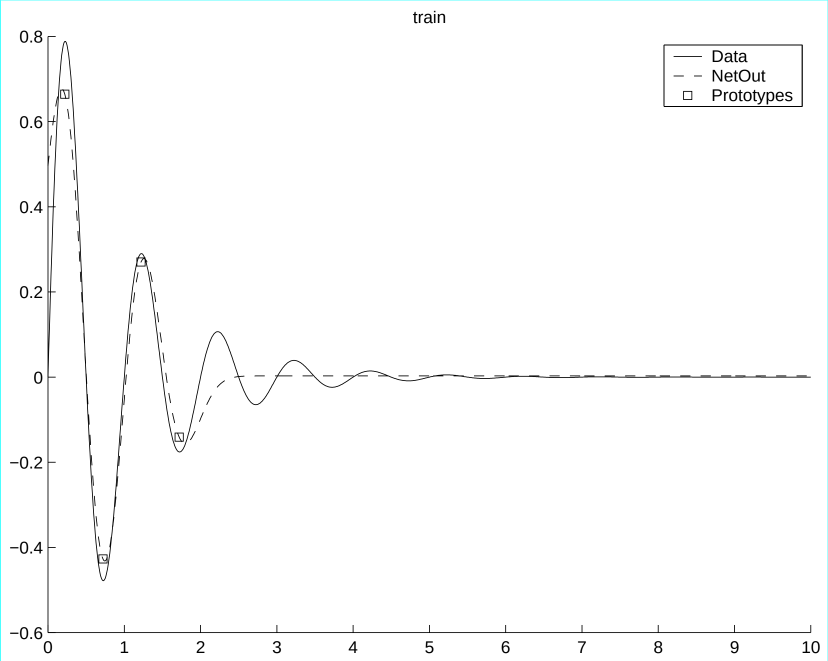
CFA Application (continued)

- The third function from CFA is a two-input data:

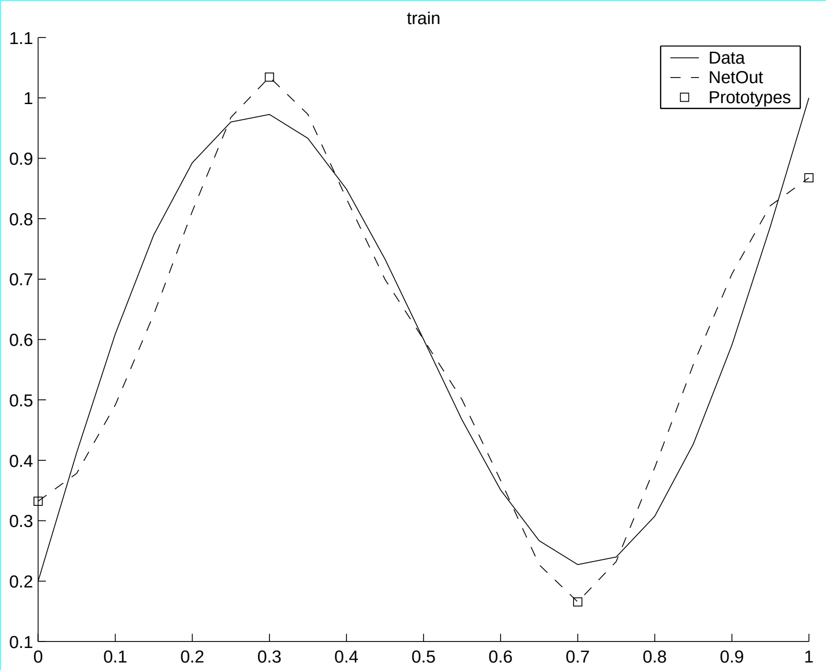
  $$f_3(x1, x2) = \frac{(x_1 - 2)(2x1 + 1)}{1 + x_1^2}\frac{(x_2 - 2)(2x_2 + 1)}{1 + x_2^2}, x_1, x_2 \in [-5, 5]$$

  where a complete set of 441 examples obtained from a grid of 21x21 points equi-distributed in the input interval defined for f_3.

Hybrid Net Regression Results



f_1 (continuous line) and the output of PRBFN (dashed line), the prototypes are shown as rectangles.



f_2 taken from CFA. The net output is in dash and the prototypes are the rectangles.

CFA Results

	f1	f2	f3
RBFN-CFA	0.952±0.001	0.380±0.035	0.926±0.008
PRBFN2	0.103±0.001	0.082±0.001	0.663±0.001

Comparison of normalized mean squared error results on three data sets The results for $RBFN{-}CFA$ are quoted from (Gonzalez et al. 2002).

Small datasets

	LogGauss	2D Sine	Elec Circ.
RBF-Reg-Tree	0.02±0.14	0.91±0.19	0.12±0.03
RBF-OLS	-	0.74±0.41	0.20±0.03
RBF-EM	0.02±0.02	0.53±0.19	0.18±0.02
PRBFN	0.02±0.02	0.53±0.21	0.15±0.03
PRBFN2	0.01±0.01	0.46±0.19	0.12±0.03

Data sets from (Orr et al, 2000). The electric circuit was taken from Friedman, (MARS got similar results).

Pumadyn Regression

- Pumadyn dynamics of puma robot arm (from DELVE).
- 8 dimension and 32 dimension input space
- Target : angular acceleration of one link.
- We used the data which is strongly corrupted by Gaussian noise
- A highly non linear problem

Methods for comparison

- **Lin-1** Linear least squares regression.
- **kNN-cv-1** KNN for regression. K is selected by CV.
- **MLP-ens-1** MLP ensembles with early stopping and conjugate gradient.
- **HME-ens-1** Hierarchical mixtures of experts. (early stopping).
- **GP-map-1** Gaussian processes for regression, using
- **MLP-MC-1** MLP (ensembles) trained by MCMC. . a maximum a-posteriori via conjugate gradient.

- **MARS3.6-bag-1** MARS with bagging.
- **PRBFN-AS-RBF** RBF with pruning.
- **PRBFN-AS-MLP** MLP with pruning.
- **PRBFN-LRT** Full PRBFN method LRT for pruning.
- **PRBFN2** PRBFN - BIC model selection and LRT pruning.

Results: 32 inputs

Training size	64	128	512	1024
Lin-1	1.98±0.25	1.20±0.05	0.89±0.02	0.86±0.02
kNN-cv-1	1.00±0.02	1.01±0.03	0.92±0.02	0.90±0.02
MLP-ens-1	1.25±0.04	1.13±0.09	0.89±0.02	0.86±0.02
HME-ens-1	1.22±0.02	1.12±0.04	0.89±0.02	0.87±0.02
GP-map-1	1.01±0.06	0.70±0.12	**0.36**±0.01	**0.35**±0.01
MLP-mc-1	0.88±0.06	0.58±0.06	0.59±0.06	**0.35**±0.01
MARS3.6-bag-1	0.93±0.06	0.53±0.03	**0.35**±0.01	**0.34**±0.01
PRBFN-AS-RBF	1.14±0.2	0.57±0.09	0.39±0.02	0.38±0.03
PRBFN-AS-MLP	1.11±0.08	0.84±0.06	0.54±0.06	0.40±0.02
PRBFN-LRT	1.45±0.2	1.14±0.09	0.55±0.05	0.44±0.03
PRBFN2	**0.75**±0.11	**0.43**±0.02	**0.37**±0.02	**0.34**±0.01

Results: 8 inputs

Training Size	64	128	512	1024
Lin-1	0.73±0.02	0.68±0.02	0.63±0.014	0.63±0.02
kNN-CV-1	0.79±0.02	0.71±0.02	0.58±0.02	0.53±0.02
MLP-ens-1	0.72±0.02	0.67±0.02	0.49±0.01	0.41±0.01
HME-ens-1	0.72±0.02	0.67±0.02	0.54±0.02	0.44±0.02
GP-map-1	**0.44**±0.03	**0.38**±0.01	**0.33**±0.01	**0.32**±0.01
MLP-MC-1	**0.45**±0.01	**0.39**±0.02	**0.32**±0.01	**0.32**±0.01
MARS3.6-bag-1	0.51±0.02	**0.38**±0.01	**0.34**±0.01	0.34±0.01
PRBFN-AS-RBF	0.51±0.03	**0.38**±0.02	**0.33**±0.01	**0.32**±0.01
PRBFN-AS-MLP	0.57±0.05	0.59±0.14	**0.33**±0.08	**0.32**±0.01
PRBFN-LRT	0.72±0.11	0.60±0.05	0.41±0.01	0.35±0.02
PRBFN2	**0.48**±0.03	**0.38**±0.01	**0.33**±0.01	**0.32**±0.01

Related work

- Hassibi et al. with Optimal Brain Surgeon
- Mackey with Bayesian inference of weights and regularization parameters
- HME Jordan and Jacob, division of input space.
- Kass and Raftery using BIC.

Summary

- Pruning removes 90% of the parameters and reduces the variance of estimator
- PRBFN is better then RBF or MLP alone.
- Bayesian techniques disadvantages: the prior distribution of parameters, but on the data tested, better than LRT.
- Determination of unit parameters, greatly reduces training time
- Unit type selection is crucial in PRBFN
- Unit selection with MAP is better than unit selection with MLE.

Classification: Initial decomposition of input space

- Breiman et al (CART 84) have used a twoing criterion for splitting region of input space

- We have adopted a similar entropy criterion which we have extended to non-parallel projections:

  $$\Delta Er(C_0) = Er(C_0) - [Er(C_1) + Er(C_2)].$$

  The definition of $Er(C_0)$ includes the empirical probability of C_0: $\hat{P}_{C_0} = |C_0|/|D|$.

Details of the (non-parallel) decomposition

Consider two subsets $V_i,\ V_j$.

Consider the two biggest class member inputs.

Let $m_i = (1/n_i)\sum_{x \in V_i} x$. be the subset mean.

Set $y_i \in \{-1, 1\}$ be the corresponding class labels.

$S_i = \sum_{x \in V_i}(x - m_i)(x - m_i)^T, S_w = S_1 + S_2$.

$w = S_w^{-1}(m1 - m2)$.

Minimize $E_w = \sum_{i=1}^{n}(w^T x_i + w_o - y_i)^2$. w.r.t w_0.

Network construction & training procedure

- Decompose the input space into homogenous regions
- Choose the appropriate unit for each specific region in input space
  - Includes determination of initial weights
- Determine network size
- Train the full network

Unit Selection

Is done via likelihood ratio between the models as before.

Initial weights

Initial weights for an RBF unit: center of the cluster.
For a projection unit, we use a linear approximation $w^T x$. Thus, we maximize

$$L(w, \alpha) = \sum_{i=1}^{N} w^T x_i$$

subject to $w^T w = 1$, which implies maximization of

$$L(w, \alpha) = \sum_{i=1}^{N} w^T x_i - \alpha(w^T w - 1),$$

$$\Rightarrow \quad w = (\sum_{i=1}^{N} x_i) / \parallel (\sum_{i=1}^{N} x_i) \parallel .$$

Network construction & training procedure

- Decompose the input space into homogenous regions
- Choose the appropriate unit for each specific region in input space
  - Includes determination of initial weights
- Determine network size
- Train the full network

Stopping criterion

N_l^i is the number of patterns from class i that are sent to the left node.
$E[N_l^i]$ is be the expected number of patterns sent due to a random split.
The χ^2 statistics is given by:

$$\chi^2 = \sum_{i=1}^{2} \frac{(N_l^i - E[N_l^i])^2}{E[N_l^i]}$$

Splitting stops when χ^2 is below a predefined confidence level.

Network construction & training procedure

- Decompose the input space into homogenous regions
- Choose the appropriate unit for each specific region in input space
  - Includes determination of initial weights
- Determine network size
- Train the full network

Full gradient descent

Gradient descent is performed on:

- The input to hidden unit weights
- The hidden to output weights
- The radii of the RBF.

Care should be taken so that the radii do not shrink to zero.

Classification results

Algorithm	Sonar	Vowel	waveform	Hepatitis	Letters
RBF-Tree	71.7±0.5	–	–	79.8±5	
RBF-OLS	82.3±2.4	51.6±2.9	83.8±0.2	82.7±3	
RBF-EM	–	48.4±2.4	83.5±0.2	77.3±3	85.49±2.0
PRBFN	91.3±2.1	67.0±2.1	85.8±0.2	82.1±4	85.5 ±1.9
PRBFN2	92.3±1.9	68.0±1.9	85.8±0.3	84.2±4	94.02 ±0.0

RBF-Tree - Orr: using regression tree for clustering.

RBF-OLS - Matlab: an incremental architecture.

RBF-EM - Bishop: EM for clustering.

PRBFN - Last years version: manual model selection.

PRBFN2 - Latest version: automatic model selection.

Summary

- Pruning removes 90% of the parameters and reduces the variance of estimator
- PRBFN is better then RBF or MLP alone.
- Bayesian techniques disadvantages: the prior distribution of parameters (but on the data tested, better than LRT.)
- Determination of unit parameters, greatly reduces training time
- Unit type selection is crucial in PRBFN
- Unit selection with MAP is better than unit selection with MLE.