The Radial Basis Function Network

March 5, 2006

Radial Basis Functions (RBFs)

- Three layers of nodes in the network.
- It firs a curve to the data using a statistical measure for the quality of fit. Unlike BP which uses stochastic approximation.
- Hidden unit activation is determined by the distance between the input vector and a prototype. Distance from a prototype is a form of clustering.

Training

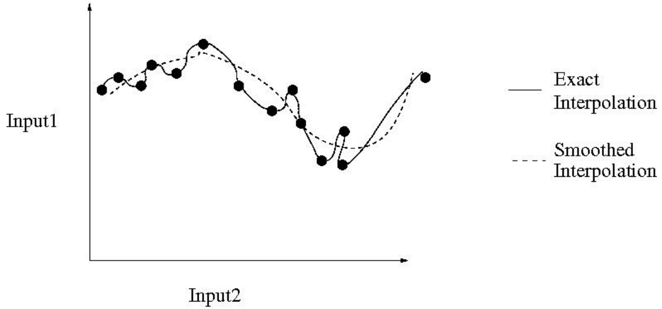
- There are two stages to training:
- 1.) Parameters of the bias functions (hidden units) are determined using unsupervised learning. The first layer of weights are trained using input data only.
- 2.) Calculate the second weight layer values. Create a linear mapping from hidden layer activations to the target output patterns.
- Both steps are relatively fast when compared to BP.

High Dimensionality Mapping

- The rationale for the two processing steps, the first is non-linear and the second linear, is attributed to Cover (1965). This work states that a pattern classification problem which is cast into a high dimensional space is more likely to be linearly separable.
- The hidden units in the RBF represent the dimensionality into which the input pattern is transformed. This is why RBF networks normally have "large" numbers of hidden units. Input patterns are transformed into a high dimensional space in the hidden layer and they are then linearly separated to classify the results.
- Also related to the dimensionality of the hidden unit space is the ability of the network to approximate a smooth input-output mapping. Higher dimensionl implies a more accurate and smoother mapping.

Exact Interpolation

- **Exact Interpolation** is a technique for interpolating a set of data points such that every input point is mapped onto a target. Every input point must appear as part of the system used to model the data without any averaging or smoothing.
- This is **not** what is desired in Radial Basis Function networks.



The RBF represented by the solid line hasn N data points and N basis functions (one per dot).

Exact Interpolation Activation

- The input vectors are labeled x^n.
- The activation function is:

$$h(x) = \sum_n w_n \theta(||x - x^n||)$$

  where $\sum_n w_n$ is a linear combination of basis functions, θ are the basis functions which are described using $||x - x^n||$ the distance between input vector x and training pattern x^n.
- Where $\theta(x)$ is often a Gaussian:

$$\theta(x) = exp(-\frac{x^2}{2\sigma^2})$$

  and σ is a smoothing parameter.
- Other functions for $\theta(x)$ can be used.

- For Exact Interpolation, only one value of σ is used.
- More than one value for σ can be used in an RBF. The quantity of smoothing may be consistent or vary over the curve. Variable smoothing values are useful for data which has more noise or variation within the samples and require more smoothing.

Problems with Exact Interpolation

- Interpolating every point is not desirable. Noisy data will often produce oscillations in the functions leading them to be unsmooth.
- A better generalization can be produced and a smoother curve can be produced if the noise is averaged.
- We would also like to reduce the number of basis functions to less than the number of input patterns.

Radial Basis Function Networks

- Changes to the Exact Interpolation algorithm which create an RBF.
- 1.) The number of basis functions M is much less than the number of input patterns N.
- 2.) The centres of the basis functions are not constrained to those given by the input vectors. Determining centres becomes part of the training process.
- 3.) Instead of using a common with parameter σ, each basis function has its own with σ_j whose value is determined during training.
- 4.) A Bias parameter is included in the linear sum. This compensates for differences between the average of the basis function activations and the average of the targets.

RBF Activation

- The equations for RBF activation is:

$$y_k(x) = \sum_{j=1}^{M} w_{kj}\theta_j(x) + w_{ko}$$

where:

$$\theta_j(x) = exp(-\frac{||x - u_j||^2}{2\sigma_j^2})$$

and x is the d dimensional input vector with elements x_i, and u_j is the vector determining the centre of the basis function θ_j.

Network Training

- There are two stages to training:
  1. determining the parameters of the basis functions through unsupervised training using only the input data set
  2. once basis functions have been determined and their parameters are set then the second layer weights w_{kj} are determined using both input and output data (hidden units are activated using an input pattern and the weights to the output layer are then modified to produce the desired output for the given input)

Output Error Calculation

- Training step 2 is the easier of the two steps as it involves solving a set of linear equations.
- The error at the output is normally calculated using a sum-of-squares function:

$$E = \frac{1}{2} \sum_n \sum_k y_k(x^n) - t_k^{n2}$$

where n is the number of input patterns, k is the sum of the values for each output node k, $y_k(x^n)$ is the achieved output for node k given input x^n and t_k^n is the desired output for node k given input n. This calculates the difference between the desired and achieved outputs for all output patterns and for all output nodes.

Output Layer Notes

- The output weight layer is normally solved using singular value decomposition.
- Other non-linear activation functions applied to the outputs and other choices for error functions are possible are possible but generally not used. This would lead to determining the second weight layer as a non-linear problem and would be a much more difficult non-linear optimization.

Basis Function Optimization (First Layer Training)

- The operations performed can be described in terms of several different techniques including:
  - regularization theory
  - noisy interpolation theory
  - kernel regression
  - function approximation
  - estimation of posterior class probabilities
- All of these methods suggest that basis functions parameters should be chosen to represent the probability density of the input data.

- The training procedure which results is an unsupervised optimization of the basis function parameters. The basis function centres u_j can be regarded as prototypes for the input vectors.
- Large amounts of labeled data can be difficult to obtain. The RBF can use large amounts of unlabeled data to train the first layer and a relatively small amount of labeled data to train the second layer.

Problems with Full Problems Spaces

- A problem occurs is the basis functions are required to **fill** the problem space. In this case, the number of basis functions increases exponentially with the dimension of the problem. This situation requires long training times and a large number of training patterns.
- The RBF performs better if it isn't required to represent the entire problem space and only needs to work in a small subset of it.
- The cost of this problem is particularly high when there are input variables which have a high variance but which have little effect in determining the output. Input variables with this property are not uncommon.
- When basis functions are selected using only input data there is no way to identify if the patterns are relevant.

Density Function Drawbacks

- The optimal choice for basis function parameters using density estimation may not be optimal for creating the mapping to the output values. This occurs when the density estimation doesn't represent the real density and centres.

Determining Basis Function Centres (u_j)

Random Selection

- Select a random subset of input vectors from the training set.
- Doesn't attempt to provide an optimal density estimation. Can require an overly large number of basis functions to achieve the desired performance.
- Often used to provide a set of starting values which can be iteratively adapted to produce a better solution.

All Data

- Use all data points as basis function centres and selectively remove centres which have the least disruption on performance.

- Both of these methods provide only the centres u_j but not the width parameter σ_J. This is often set equal to a multiple of the average distance between the centres. This causes the functions to overlap and therefore provides a relatively smooth representation of the distribution of the training data. As all widths are equal, this may not be the best solution.
- These methods do no supply optimal parameters but they are very fast.

Determining Basis Function Centres (u_j)

Orthogonal Least Squares

- Involves starting with one basis function and adding more which are selected to create the greatest reduction of error.
- Basis functions centres are taken from data points. Centres are selected by constructing a set of orthogonal vectors in the space spanned by the hidden unit activations for each training pattern.
- This allows the choice of the next centre which will produces the greatest reduction in error. Orthogonal vectors are those which are most "different" from those already chosen which should reduce error by the largest amount.
- If the algorithm is allowed to run to completion then it will select all data points. Good generalization requires that it stop before this occurs.

Determining Basis Function Centres (u_j)

Clustering Algorithms

- Instead of selecting a subset of data points for the centres we can find a set of clusters which better represent the distribution.
- Data points for centres may not be the best choice. Cluster centres are not required to be data points.
- K-means clustering provides K centres where K must be decided in advance. The algorithm divides the data into K subsets so that the distance between cluster centres and points is minimized.
- Selecting a value for K may not be obvious.
- An alternative to K-means clustering is to use a neural network which performs a similar function. The Kohonen Self-Organizing Feature map generates a set of prototypes vectors which represent input vector-space values. These prototypes can be used to create basis function centres.

Determining Basis Function Centres (u_j)

Gaussian Mixture Models

- The problem is one of density estimation where the basis functions are the components of a mixture of density model whose parameters are optimized by maximum likelihood.
- A purely statistical technique using no heuristics.
- Density is modeled using:

$$p(x) = \sum_{j=1}^{M} P(j)\theta_j(x)$$

where $P(j)$ is the prior probabilities for data points to have been generated by the j^{th} component and $\theta_j(x)$ are the basis functions.

- The likelihood function is maximized with respect to $P(j)$ and the parameters of $\theta_j(x)$. This is done by computing the derivatives of the likelihood function with respect to the parameters and then optimizing them.
- Parameters can also be found using re-estimation methods based on expected maximization procedures.

Comparing RBFs and Back-Propagation (Multi-Layer Perceptron)

- Both approximate arbitrary non-linear functional mappings of multidimensional spaces.
- Mappings are created through combinations of multiple functions.
- Hidden units in BP are the sum of inputs transformed through a threshold function. Hidden unit activation in an RBF uses a distance to a prototype followed by a local transformation.
- BP uses a distributed representation where many hidden units contribute to a solution for a given input. The training process is highly non-linear and there are problems with local solutions. This can lead to slow convergence. RBFs use localized basis functions and typically only a few hidden units have significant activations.

- BP networks can have multiple layers of weights which can vary in many ways. RBFs have a simple three layer structure which does not change.
- BP generally uses one training method where RBFs can determine the basis functions through many methods.
- All parameters in BP are determined simultaneously using a single global supervised training strategy. RBFs are training in two stages, the first of which is unsupervised and the second is a linear supervised method.