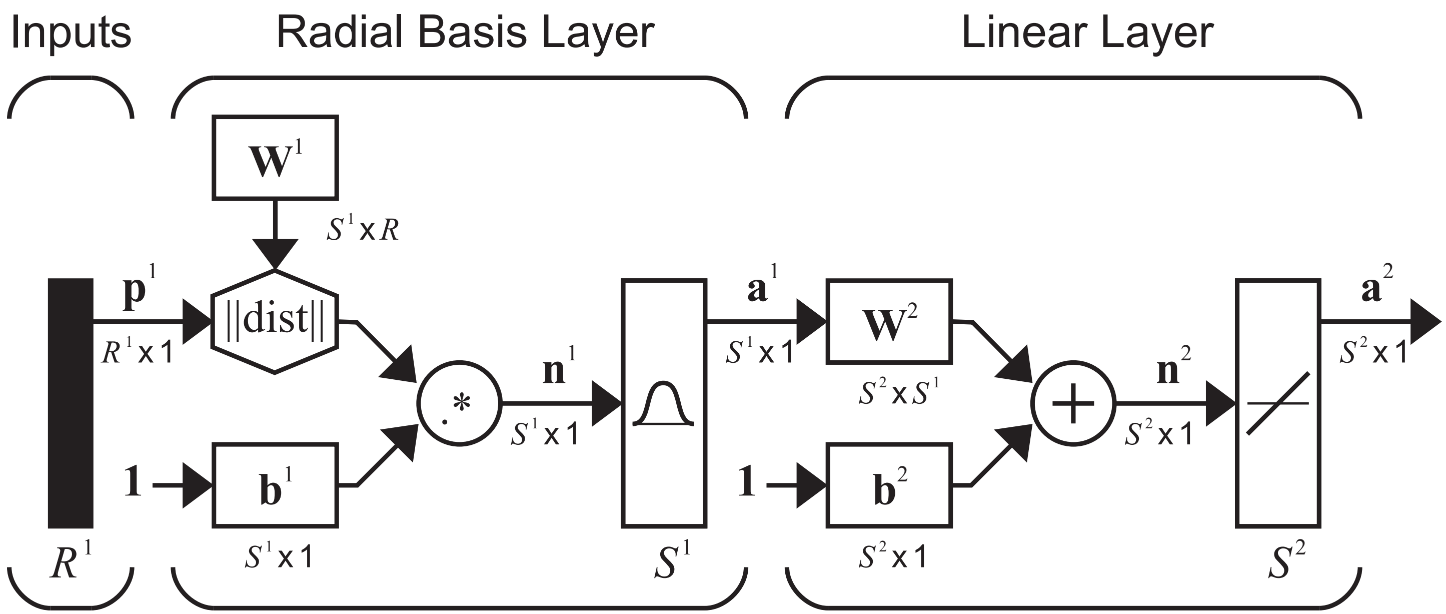
Radial Basis Networks

Radial Basis Network

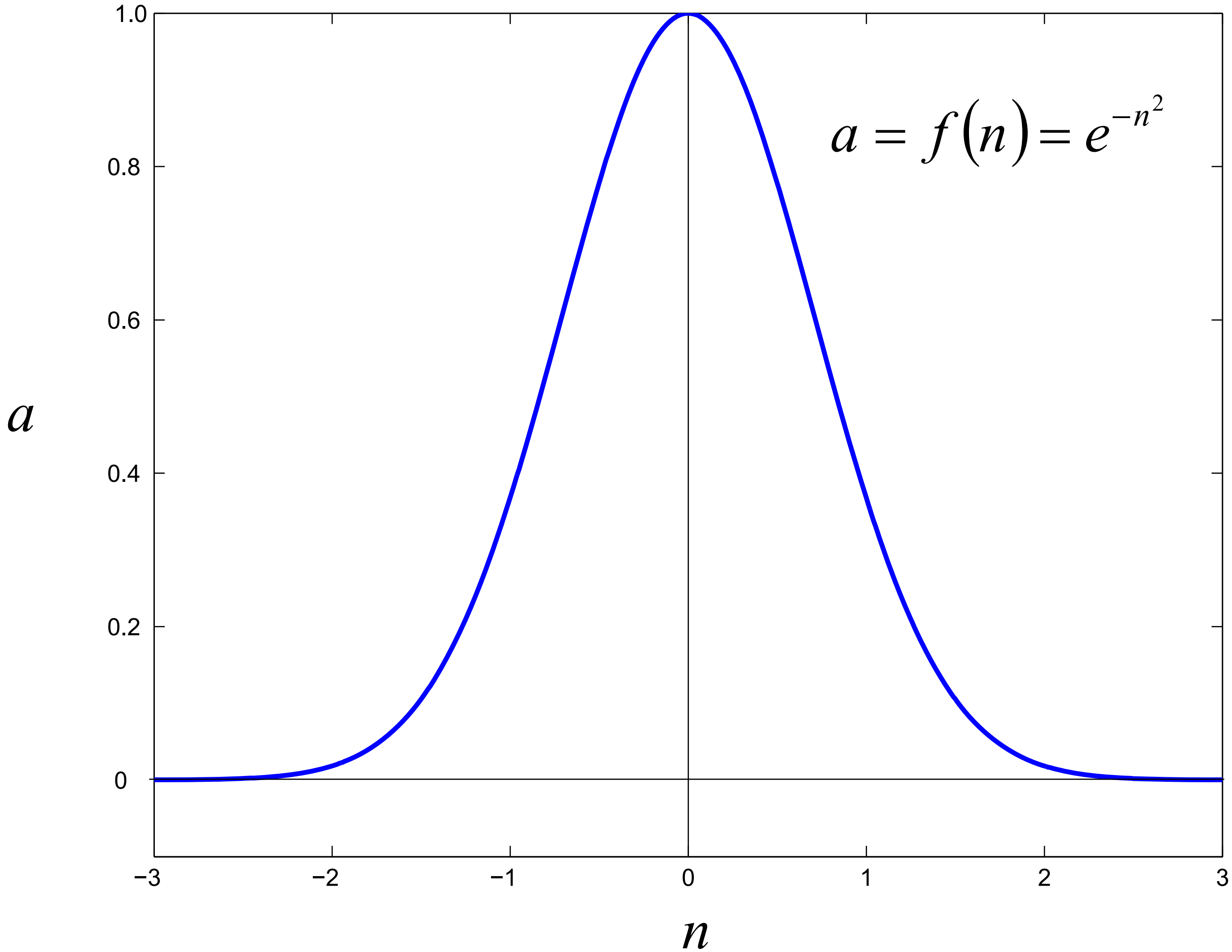


$a^1_i = radbas(\|_i\mathbf{w}^1\text{-}\mathbf{p}\|b^1_i)$ $\qquad$ $\mathbf{a}^2 = \mathbf{W}^2\mathbf{a}^1 + \mathbf{b}^2$

$$n_i^1 = \left\| \mathbf{p} - {}_i\mathbf{w}^1 \right\| b_i^1 \qquad b = 1/\left(\sigma\sqrt{2}\right) \qquad a = f(n) = e^{-n^2}$$

The first layer weight vectors ${}_i\mathbf{w}^1$ are called "centers" of the basis functions.

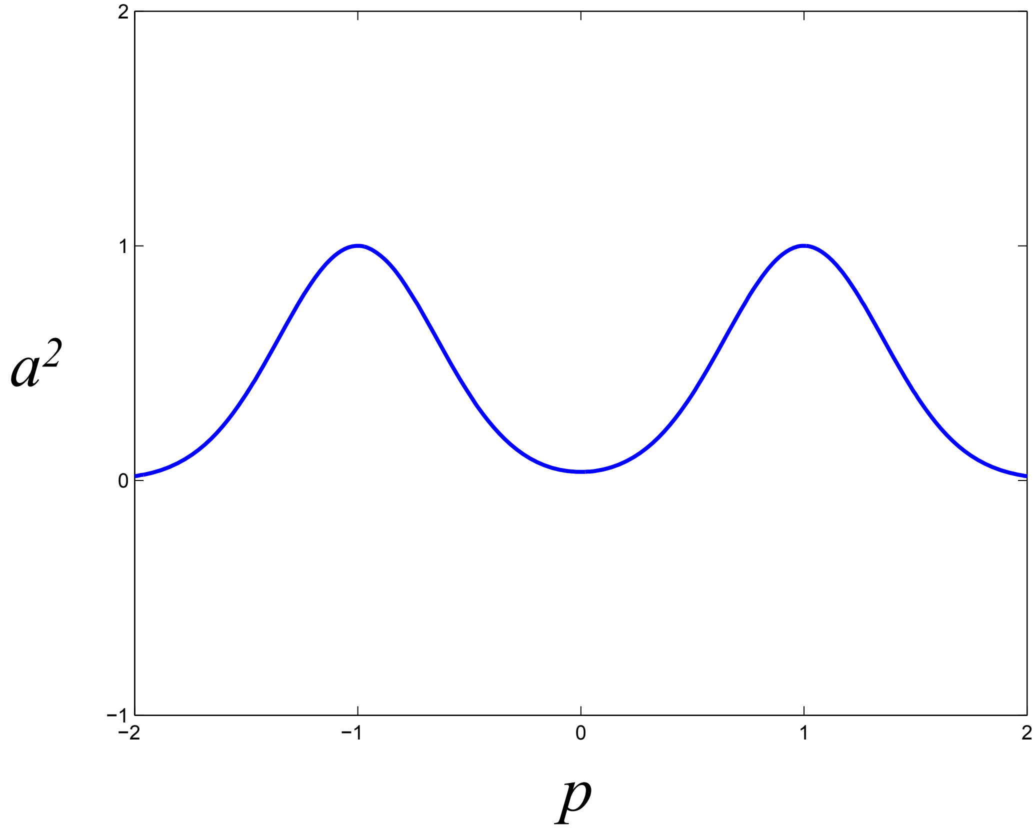
Gaussian Transfer Function (Local)



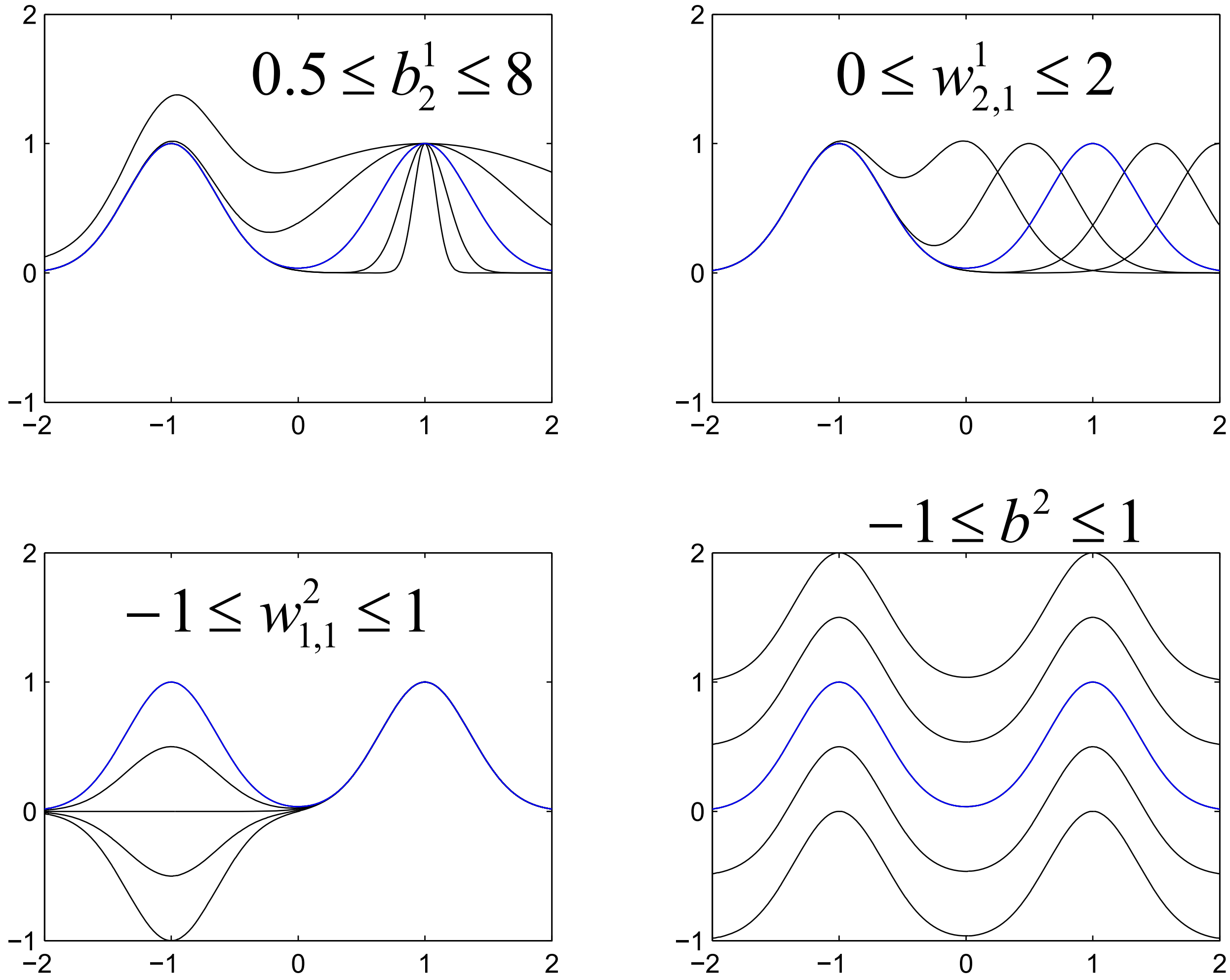
Example Network Function

$$w^1_{1,1} = -1,\ w^1_{2,1} = 1,\ b^1_1 = 2,\ b^1_2 = 2$$

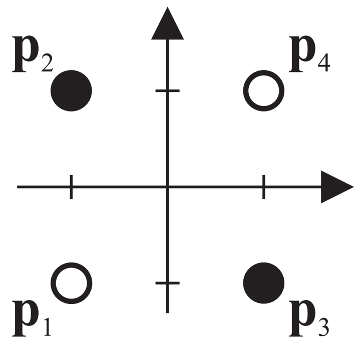
$$w^2_{1,1} = 1,\ w^2_{1,2} = 1,\ b^2 = 0$$



Parameter Variations



Pattern Recognition Problem

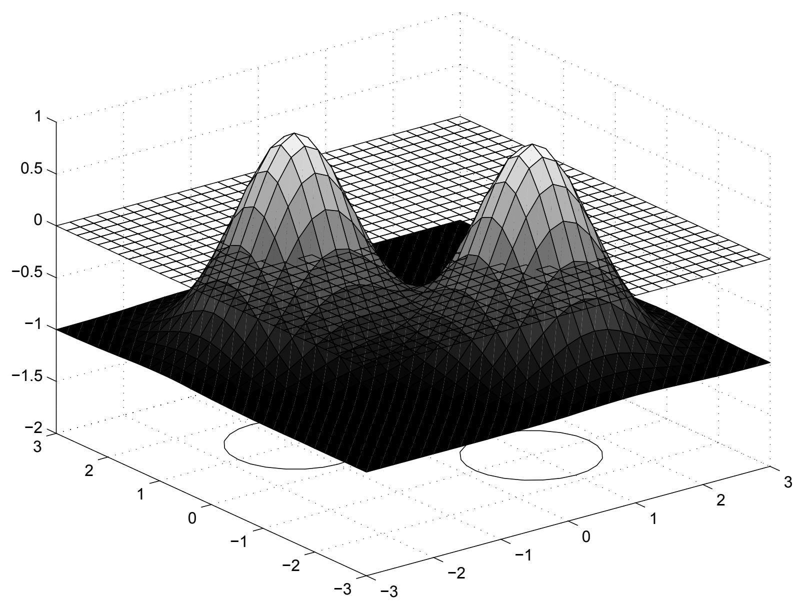


$$\text{Category 1}: \left\{ \mathbf{p}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \mathbf{p}_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \quad \text{Category 2}: \left\{ \mathbf{p}_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{p}_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$$

Radial Basis Solution

Choose centers at $\mathbf{p}_2$ and $\mathbf{p}_3$:

$$\mathbf{W}^1 = \begin{bmatrix} \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$



Choose bias to be 1:

$$\mathbf{b}^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

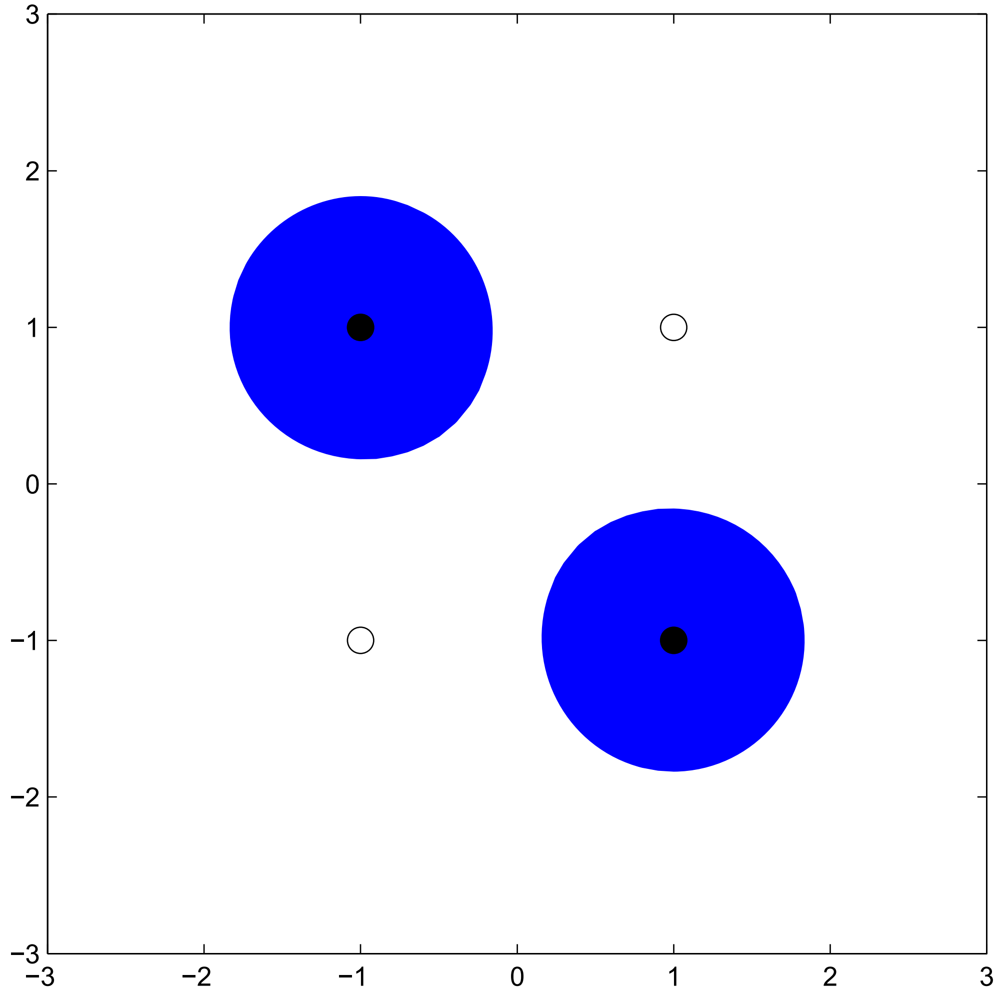
This will cause the following reduction in the basis functions where they meet:

$$a = e^{-n^2} = e^{-\left(1 \cdot \sqrt{2}\right)^2} = e^{-2} = 0.1353$$

Choose the second layer bias to produce negative outputs, unless we are near $\mathbf{p}_2$ and $\mathbf{p}_3$. Choose second layer weights so that output moves above 0 near $\mathbf{p}_2$ and $\mathbf{p}_3$.

$$\mathbf{W}^2 = \begin{bmatrix} 2 & 2 \end{bmatrix}, b^2 = \begin{bmatrix} -1 \end{bmatrix}$$

Final Decision Regions



Global Versus Local

- Multilayer networks create a distributed representation.
  - All sigmoid or linear transfer functions overlap in their activity.
- Radial basis networks create local representations.
  - Each basis function is only active over a small region.
- The global approach requires fewer neurons. The local approach is susceptible to the "curse of dimensionality."
- The local approach leads to faster training and is suitable for adaptive methods.

Radial Basis Training

- Radial basis network training generally consists of two stages.
- During the first stage, the weights and biases in the first layer are set. This can involve unsupervised training or even random selection of the weights.
- The weights and biases in the second layer are found during the second stage. This usually involves linear least squares, or LMS for adaptive training.
- Backpropagation (gradient-based) algorithms can also be used for radial basis networks.

Assume Fixed First Layer

We begin with the case where the first layer weights (centers) are fixed. Assume they are set on a grid, or randomly set. For random weights, the bias can be

$$b_i^1 = \frac{\sqrt{S^1}}{d_{\max}}$$

The training data is given by

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

With first layer weights and biases fixed, the first layer output can be computed:

$$n_{i,q}^1 = \left\| p_q - {}_i w^1 \right\| b_i^1 \qquad \mathbf{a}_q^1 = \mathbf{radbas}\left(\mathbf{n}_q^1\right)$$

This provides a training set for the second layer:

$$\{\mathbf{a}_1^1, \mathbf{t}_1\}, \{\mathbf{a}_2^1, \mathbf{t}_2\}, \ldots, \{\mathbf{a}_Q^1, \mathbf{t}_Q\}$$

17 Linear Least Squares (2nd Layer)

$$\mathbf{a}^2 = \mathbf{W}^2\mathbf{a}^1 + \mathbf{b}^2 \qquad F(\mathbf{x}) = \sum_{q=1}^{Q}\left(\mathbf{t}_q - \mathbf{a}_q^2\right)^T\left(\mathbf{t}_q - \mathbf{a}_q^2\right)$$

$$\mathbf{x} = \begin{bmatrix} {}_1\mathbf{w}^2 \\ b^2 \end{bmatrix} \qquad \mathbf{z}_q = \begin{bmatrix} \mathbf{a}_q^1 \\ 1 \end{bmatrix}$$

$$a_q^2 = \left({}_1w^2\right)^T a_q^1 + b^2 = \mathbf{x}^T\mathbf{z}_q$$

$$F(\mathbf{x}) = \sum_{q=1}^{Q}\left(\mathbf{t}_q - \mathbf{x}^T\mathbf{z}_q\right)^T\left(\mathbf{t}_q - \mathbf{x}^T\mathbf{z}_q\right)$$

17

Matrix Form

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_Q \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} {}_1\mathbf{u}^T \\ {}_2\mathbf{u}^T \\ \vdots \\ {}_Q\mathbf{u}^T \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_Q^T \end{bmatrix} \qquad \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_Q \end{bmatrix}$$

$$\mathbf{e} = \mathbf{t} - \mathbf{U}\mathbf{x} \qquad F(\mathbf{x}) = (\mathbf{t} - \mathbf{U}\mathbf{x})^T(\mathbf{t} - \mathbf{U}\mathbf{x})$$

$$F(\mathbf{x}) = (\mathbf{t} - \mathbf{U}\mathbf{x})^T(\mathbf{t} - \mathbf{U}\mathbf{x}) + \rho\sum_{i=1}^{n} x_i^2 = (\mathbf{t} - \mathbf{U}\mathbf{x})^T(\mathbf{t} - \mathbf{U}\mathbf{x}) + \rho\mathbf{x}^T\mathbf{x}$$

$$= \mathbf{t}^T\mathbf{t} - 2\mathbf{t}^T\mathbf{U}\mathbf{x} + \mathbf{x}^T\mathbf{U}^T\mathbf{U}\mathbf{x} + \rho\mathbf{x}^T\mathbf{x}$$

$$= \mathbf{t}^T\mathbf{t} - 2\mathbf{t}^T\mathbf{U}\mathbf{x} + \mathbf{x}^T[\mathbf{U}^T\mathbf{U} + \rho\mathbf{I}]\mathbf{x}$$

17

Linear Least Squares Solution

$$F(\mathbf{x}) = \mathbf{t}^T\mathbf{t} - 2\mathbf{t}^T\mathbf{U}\mathbf{x} + \mathbf{x}^T\left[\mathbf{U}^T\mathbf{U} + \rho\mathbf{I}\right]\mathbf{x}$$

$$= c + \mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} \qquad \text{(Quadratic Function)}$$

$$\nabla F(\mathbf{x}) = \nabla\left(c + \mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}\right) = \mathbf{d} + \mathbf{A}\mathbf{x}$$

$$= -2\mathbf{U}^T\mathbf{t} + 2\left[\mathbf{U}^T\mathbf{U} + \rho\mathbf{I}\right]\mathbf{x} = 0$$

$$\left[\mathbf{U}^T\mathbf{U} + \rho\mathbf{I}\right]\mathbf{x}^* = \mathbf{U}^T\mathbf{t}$$

Example (1)

$$g(p) = 1 + \sin\left(\frac{\pi}{4} p\right) \text{ for } -2 \le p \le 2$$

$$p = \{-2, -1.2, -0.4, 0.4, 1.2, 2\}$$

$$t = \{0, 0.19, 0.69, 1.3, 1.8, 2\}$$

$$\mathbf{W}^1 = \begin{bmatrix} -2 \\ 0 \\ 2 \end{bmatrix}, \mathbf{b}^1 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

Example (2)

$$n_{i,q}^1 = \| p_q - {}_iw^1 \| b_i^1 \qquad \mathbf{a}_q^1 = \mathbf{radbas}(\mathbf{n}_q^1)$$

$$\mathbf{a}^1 = \left\{ \begin{bmatrix} 1 \\ 0.368 \\ 0.018 \end{bmatrix}, \begin{bmatrix} 0.852 \\ 0.698 \\ 0.077 \end{bmatrix}, \begin{bmatrix} 0.527 \\ 0.961 \\ 0.237 \end{bmatrix}, \begin{bmatrix} 0.237 \\ 0.961 \\ 0.527 \end{bmatrix}, \begin{bmatrix} 0.077 \\ 0.698 \\ 0.852 \end{bmatrix}, \begin{bmatrix} 0.018 \\ 0.368 \\ 1 \end{bmatrix} \right\}$$

$$\mathbf{U}^T = \begin{bmatrix} 1 & 0.852 & 0.527 & 0.237 & 0.077 & 0.018 \\ 0.368 & 0.698 & 0.961 & 0.961 & 0.698 & 0.368 \\ 0.018 & 0.077 & 0.237 & 0.527 & 0.852 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{t}^T = \begin{bmatrix} 0 & 0.19 & 0.69 & 1.3 & 1.8 & 2 \end{bmatrix}$$

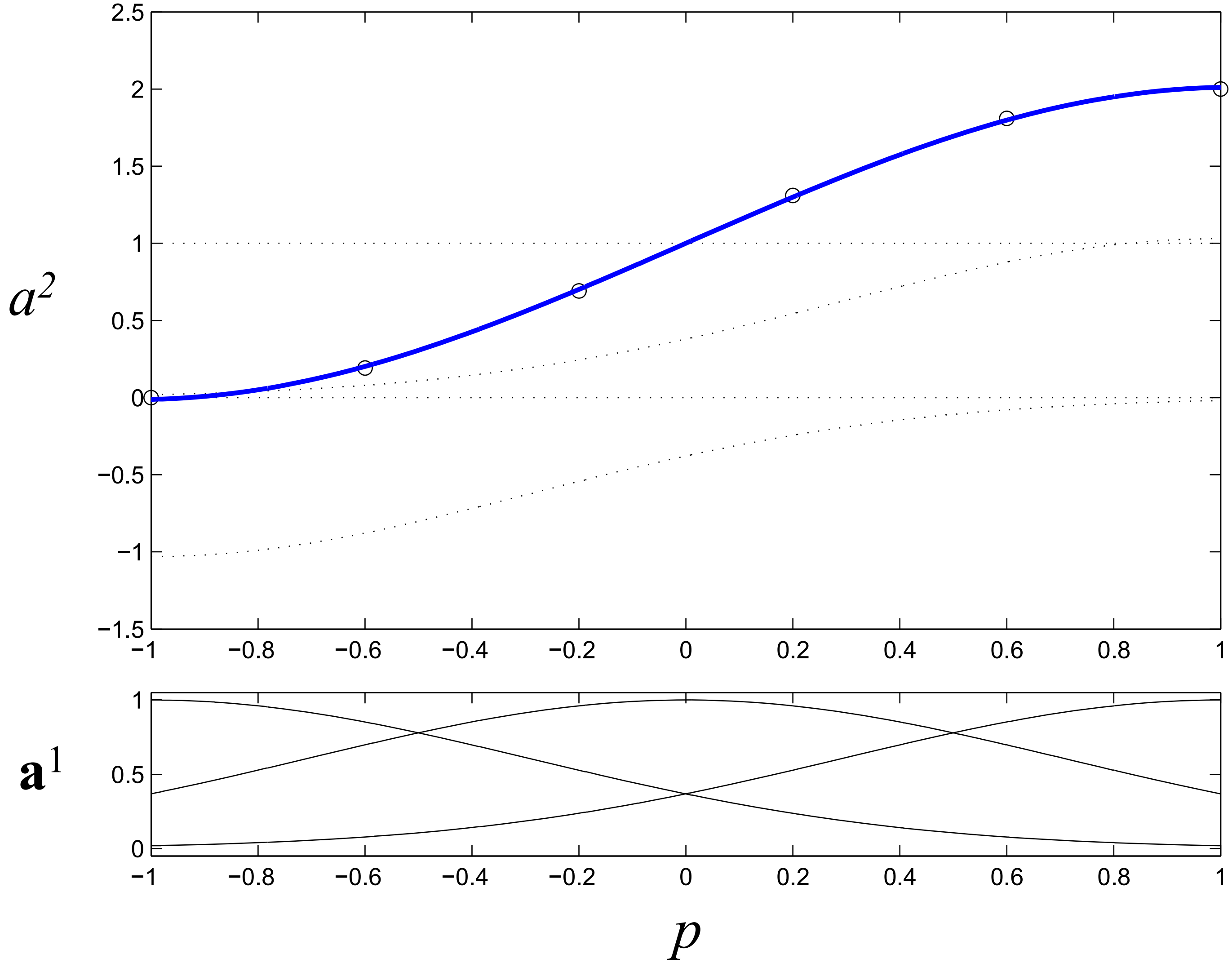
Example (3)

$$\mathbf{x}^* = \left[\mathbf{U}^T\mathbf{U} + \rho\mathbf{I}\right]^{-1}\mathbf{U}^T\mathbf{t}$$

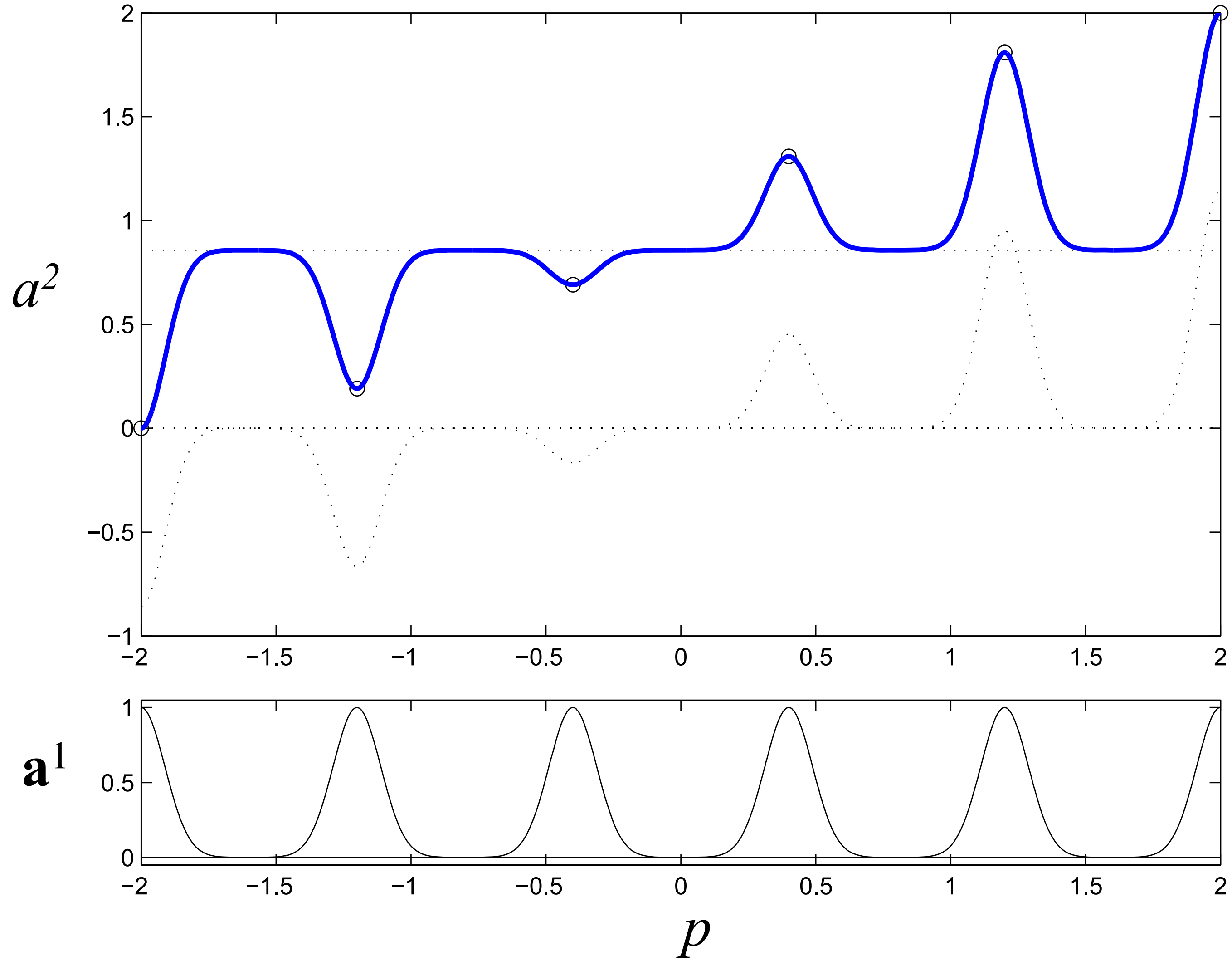
$$\mathbf{x}^* = \begin{bmatrix} 2.07 & 1.76 & 0.42 & 2.71 \\ 1.76 & 3.09 & 1.76 & 4.05 \\ 0.42 & 1.76 & 2.07 & 2.71 \\ 2.71 & 4.05 & 2.71 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 1.01 \\ 4.05 \\ 4.41 \\ 6 \end{bmatrix} = \begin{bmatrix} -1.03 \\ 0 \\ 1.03 \\ 1 \end{bmatrix}$$

$$\mathbf{W}^2 = \begin{bmatrix} -1.03 & 0 & 1.03 \end{bmatrix} \qquad \mathbf{b}^2 = \begin{bmatrix} 1 \end{bmatrix}$$

Example (4)



Bias Too Large



$$\mathbf{b}^1 = \begin{bmatrix} 8 \\ 8 \\ 8 \end{bmatrix}$$

17

Subset Selection

- Given a set of potential first layer weights (centers), which combination should we use?
- An exhaustive search is too expensive.
- Forward selection begins with an empty set and adds centers one at a time.
- Backward elimination begins by using all of the potential centers and then removes them one at a time.
- There are other combinations of the forward and backward methods.
- We will concentrate on one forward selection method, called Orthogonal Least Squares.

Forward Selection

$$\mathbf{t} = \mathbf{U}\mathbf{x} + \mathbf{e}$$

$$\mathbf{U} = \begin{bmatrix} {}_1\mathbf{u}^T \\ {}_2\mathbf{u}^T \\ \vdots \\ {}_Q\mathbf{u}^T \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_Q^T \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{bmatrix} \qquad n = S^1 + 1$$

- There will be one row of **U** for each input/target pair.
- If we consider all input vectors as potential centers, there will be one first-layer neuron for each input vector: $n=Q+1$.
- In this case, the columns of **U** represent the potential centers.
- We will start with zero centers selected, and at each step we will add the center (or column of **U**) which produces the largest reduction in squared error.

17

Orthogonalize the Columns

$$\mathbf{U} = \mathbf{M}\mathbf{R}$$

$$\mathbf{R} = \begin{bmatrix} 1 & r_{1,2} & r_{1,3} & \cdots & r_{1,n} \\ 0 & 1 & r_{2,3} & \cdots & r_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & r_{n-1,n} \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$\mathbf{M}^T\mathbf{M} = \mathbf{V} = \begin{bmatrix} v_{1,1} & 0 & \cdots & 0 \\ 0 & v_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & v_{n,n} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1^T\mathbf{m}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{m}_2^T\mathbf{m}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{m}_n^T\mathbf{m}_n \end{bmatrix}$$

17 Orthogonalized Least Squares

$$\mathbf{t} = \mathbf{M}\mathbf{R}\mathbf{x} + \mathbf{e} = \mathbf{M}\mathbf{h} + \mathbf{e}$$

$$\mathbf{h} = \mathbf{R}\mathbf{x}$$

$$\mathbf{h}^{*} = \left[\mathbf{M}^{T}\mathbf{M}\right]^{-1}\mathbf{M}^{T}\mathbf{t} = \mathbf{V}^{-1}\mathbf{M}^{T}\mathbf{t}$$

$$h_i^{*} = \frac{\mathbf{m}_i^{T}\mathbf{t}}{v_{i,i}} = \frac{\mathbf{m}_i^{T}\mathbf{t}}{\mathbf{m}_i^{T}\mathbf{m}_i}$$

17 Gram-Schmidt Orthogonalization

$$\mathbf{m}_1 = \mathbf{u}_1$$

$$\mathbf{m}_k = \mathbf{u}_k - \sum_{i=1}^{k-1} r_{i,k} \mathbf{m}_i$$

$$r_{i,k} = \frac{\mathbf{m}_i^T \mathbf{u}_k}{\mathbf{m}_i^T \mathbf{m}_i}, \quad i = 1, \ldots, k-1$$

Incremental Error

The total squared value is:

$$\mathbf{t}^T\mathbf{t} = [\mathbf{M}\mathbf{h} + \mathbf{e}]^T[\mathbf{M}\mathbf{h} + \mathbf{e}] = \mathbf{h}^T\mathbf{M}^T\mathbf{M}\mathbf{h} + \mathbf{e}^T\mathbf{M}\mathbf{h} + \mathbf{h}^T\mathbf{M}^T\mathbf{e} + \mathbf{e}^T\mathbf{e}$$

$$\mathbf{e}^T\mathbf{M}\mathbf{h} = [\mathbf{t} - \mathbf{M}\mathbf{h}]^T\mathbf{M}\mathbf{h} = \mathbf{t}^T\mathbf{M}\mathbf{h} - \mathbf{h}^T\mathbf{M}^T\mathbf{M}\mathbf{h}$$

$$\mathbf{h}^* = \mathbf{V}^{-1}\mathbf{M}^T\mathbf{t} \quad \Longrightarrow \quad \mathbf{e}^T\mathbf{M}\mathbf{h}^* = \mathbf{t}^T\mathbf{M}\mathbf{h}^* - \mathbf{t}^T\mathbf{M}\mathbf{V}^{-1}\mathbf{M}^T\mathbf{M}\mathbf{h}^* = \mathbf{0}$$

$$\mathbf{t}^T\mathbf{t} = \mathbf{h}^T\mathbf{M}^T\mathbf{M}\mathbf{h} + \mathbf{e}^T\mathbf{e} = \sum_{i=1}^{n} h_i^2\mathbf{m}_i^T\mathbf{m}_i + \mathbf{e}^T\mathbf{e}$$

Therefore, basis function *i* contributes the following to the squared value:

$$h_i^2\mathbf{m}_i^T\mathbf{m}_i$$

Normalized error contribution: $$o_i = \frac{h_i^2\mathbf{m}_i^T\mathbf{m}_i}{\mathbf{t}^T\mathbf{t}}$$

OLS Algorithm

First Step ($k = 1$):

$$\mathbf{m}_1^{(i)} = \mathbf{u}_i, \quad i = 1, \ldots, Q \qquad h_1^{(i)} = \frac{\mathbf{m}_1^{(i)T}\mathbf{t}}{\mathbf{m}_1^{(i)T}\mathbf{m}_1^{(i)}}$$

$$o_1^i = \frac{\left(h_1^{(i)}\right)^2 \mathbf{m}_1^{(i)T}\mathbf{m}_1^{(i)}}{\mathbf{t}^T\mathbf{t}} \qquad o_1 = o_1^{(i_1)} = \max\left\{o_1^{(i)}\right\} \qquad \mathbf{m}_1 = \mathbf{m}_1^{(i_1)} = \mathbf{u}_{i_1}$$

For $i = 1, \ldots, Q, \quad i \neq i_1, i \neq i_2, \ldots, i \neq i_{k-1}$

$$r_{j,k}^{(i)} = \frac{\mathbf{m}_j^T\mathbf{u}_i}{\mathbf{m}_j^T\mathbf{m}_j}, \quad j = 1, \ldots, k-1 \qquad \mathbf{m}_k^{(i)} = \mathbf{u}_i - \sum_{j=1}^{k-1} r_{j,k}^{(i)}\mathbf{m}_j$$

$$h_k^{(i)} = \frac{\mathbf{m}_k^{(i)T}\mathbf{t}}{\mathbf{m}_k^{(i)T}\mathbf{m}_k^{(i)}} \qquad o_k^i = \frac{\left(h_k^{(i)}\right)^2 \mathbf{m}_k^{(i)T}\mathbf{m}_k^{(i)}}{\mathbf{t}^T\mathbf{t}} \qquad o_k = o_k^{(i_k)} = \max\left\{o_k^{(i)}\right\}$$

$$r_{j,k} = r_{j,k}^{(i_k)}, \quad j = 1, \ldots, k-1 \qquad \mathbf{m}_k = \mathbf{m}_k^{(i_k)}$$

Stopping Criteria

17

$$1 - \sum_{j=1}^{k} o_j < \delta$$

To convert to original weights:

$$x_n = h_n, \quad x_k = h_k - \sum_{j=k+1}^{n} r_{j,k} x_j$$

17 Competitive Learning for First Layer

- Cluster the input space using a competitive layer (or Feature Map).
- Use the cluster centers as basis function centers.
- The bias can be computed from the variation in each cluster:

$$dist_i = \frac{1}{n_c}\left(\sum_{j=1}^{n_c}\left\|\mathbf{p}_j^i - {}_i\mathbf{w}^1\right\|^2\right)^{\frac{1}{2}}$$

$$b_i^1 = \frac{1}{\sqrt{2}dist_i}$$

Backpropagation

$$n_i^1 = \|\mathbf{p} - {}_i\mathbf{w}^1\| b_i^1 = b_i^1 \sqrt{\sum_{j=1}^{S^1} \left(p_j - w_{i,j}^1\right)^2}$$

$$\frac{\partial n_i^1}{\partial w_{i,j}^1} = \frac{b_i^1 \frac{1}{2}}{\sqrt{\sum_{j=1}^{S^1} \left(p_j - w_{i,j}^1\right)^2}} 2\left(p_j - w_{i,j}^1\right)(-1) = \frac{b_i^1\left(w_{i,j}^1 - p_j\right)}{\|\mathbf{p} - {}_i\mathbf{w}^1\|}$$

$$\frac{\partial n_i^1}{\partial b_i^1} = \|\mathbf{p} - {}_i\mathbf{w}^1\|$$

$$\frac{\partial \hat{F}}{\partial w_{i,j}^1} = s_i^1 \frac{b_i^1\left(w_{i,j}^1 - p_j\right)}{\|\mathbf{p} - {}_i\mathbf{w}^1\|} \qquad \frac{\partial \hat{F}}{\partial b_i^1} = s_i^1 \|\mathbf{p} - {}_i\mathbf{w}^1\|$$