Function Approximation
Case Study:
Smart Sensor
Smart Sensor Diagram

[Diagram showing a light source casting a shadow on two vertical walls, labeled as $v_1$ and $v_2$. The shadow's position is indicated by $y$.]
Collected Data

Raw Data

Normalized Data

\[ p = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \]

\[ t = y \]
Network Architecture

**Inputs**
- $p^1_{2 \times 1}$
- $b^1_{S^1 \times 1}$
- $a^1 = \text{tansig}(W^1 p + b^1)$

**Tan-Sigmoid Layer**
- $n^1_{S^1 \times 1}$
- $a^1_{S^1 \times 1}$

**Linear Layer**
- $n^2_{1 \times S^1}$
- $a^2 = \text{purelin}(W^2 a^1 + b^2)$
- $a^2_{1 \times 1}$
Training Performance ($S^1=10$)

Final performance using five random initial weights.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.121e-003</td>
<td>8.313e-004</td>
<td>1.068e-003</td>
<td>8.672e-004</td>
<td>8.271e-004</td>
<td></td>
</tr>
</tbody>
</table>
Number of Hidden Layer Neurons

Sum Squared Error

<table>
<thead>
<tr>
<th>$S^l = 3$</th>
<th>$S^l = 5$</th>
<th>$S^l = 8$</th>
<th>$S^l = 10$</th>
<th>$S^l = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.406e-003</td>
<td>9.227e-004</td>
<td>8.088e-004</td>
<td>8.672e-004</td>
<td>8.096e-004</td>
</tr>
</tbody>
</table>

After a sufficient number of neurons is reached (~5) the error does not go down, if Bayesian regularization is used.
Scatter Plots

Training

Testing
\[ a = (a^n + 1) \cdot \frac{(t_{\text{max}} - t_{\text{min}})}{2} + t_{\text{min}} \]
Trained Network Response