How many hidden nodes?

- Too few hidden nodes ⇒ cannot solve task
- Too many hidden nodes ⇒ overfit data, insufficient generalization
  - Start small
  - increase number of hidden node if unable to learn
    (but not before trying different initial conditions)
How much data?

• Use enough data to force generalization
• Too few data points $\Rightarrow$ overfitting, insufficient generalization

<table>
<thead>
<tr>
<th>high</th>
<th>more</th>
<th>more</th>
</tr>
</thead>
<tbody>
<tr>
<td>concept</td>
<td>$\Rightarrow$ weights</td>
<td>$\Rightarrow$ data points</td>
</tr>
<tr>
<td>complexity</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In general: $|\text{training set}| \gg N$
Overtraining can lead to overfitting

Minimizing error on training set can lead to “overfitting”.

This can prevent generalization
Scaling problem

- Many important problems require solutions to large-scale problems.
  Vision: about 100,000 nameable icons
  How do we train such large systems
- Classes are typically presented to a system over its life-time
  all input classes are not known in advance
  all data is not available at once

Scaling problem is particularly critical for BP nets:

- BP learning is slow:
- *catastrophic interference*:

When a net trained to learn problem $A$ is subsequently trained to solve problem $B$, it forgets the solution to problem $A$. 
Controlling overfitting/generalization

During training periodically evaluate performance on validation set.

Select net that gives lowest error on validation set

Several ways of choosing validation sets...