Parameter Tuning

1. Initial weights?
2. Learning rate?
3. Momentum?
4. Generalization?
5. # Hidden nodes?
6. Recurrent links?
Over-Training Prevention

• Running too many epochs can result in over-fitting.

• Keep a hold-out validation set and test accuracy on it after every epoch. Stop training when additional epochs actually increase validation error.

• To avoid losing training data for validation:
  – Use internal 10-fold CV on the training set to compute the average number of epochs that maximizes generalization accuracy.
  – Train final network on complete training set for this many epochs.
Determining the Best Number of Hidden Units

- Too few hidden units prevents the network from adequately fitting the data.
- Too many hidden units can result in over-fitting.
- Use internal cross-validation to empirically determine an optimal number of hidden units.

![Graph showing error vs. number of hidden units for training and test data](image)
Successful Applications

• Text to Speech (NetTalk)
• Fraud detection
• Financial Applications
  – HNC (eventually bought by Fair Isaac)
• Chemical Plant Control
  – Pavillion Technologies
• Automated Vehicles
• Game Playing
  – Neurogammon
• Handwriting recognition
Issues in Neural Nets

• More efficient training methods:
  – Quickprop
  – Conjugate gradient (exploits 2\textsuperscript{nd} derivative)

• Learning the proper network architecture:
  – Grow network until able to fit data
    • Cascade Correlation
    • Upstart
  – Shrink large network until unable to fit data
    • Optimal Brain Damage

• Recurrent networks that use feedback and can learn finite state machines with “backpropagation through time.”
Issues in Neural Nets (cont.)

- More biologically plausible learning algorithms based on Hebbian learning.
- Unsupervised Learning
  - Self-Organizing Feature Maps (SOMs)
- Reinforcement Learning
  - Frequently used as function approximators for learning value functions.
- Neuroevolution
- Deep Nets
Axioms of Probability Theory

• All probabilities between 0 and 1
  \[ 0 \leq P(A) \leq 1 \]

• True proposition has probability 1, false has probability 0.
  \[ P(\text{true}) = 1 \quad P(\text{false}) = 0. \]

• The probability of disjunction is:
  \[ P(A \lor B) = P(A) + P(B) - P(A \land B) \]
Conditional Probability

- $P(A \mid B)$ is the probability of $A$ given $B$
- Assumes that $B$ is all and only information known.
- Defined by:

$$P(A \mid B) = \frac{P(A \land B)}{P(B)}$$
Independence

• A and B are independent iff:

\[ P(A | B) = P(A) \]
\[ P(B | A) = P(B) \]

These two constraints are logically equivalent

• Therefore, if A and B are independent:

\[ P(A | B) = \frac{P(A \land B)}{P(B)} = P(A) \]

\[ P(A \land B) = P(A)P(B) \]
The joint probability distribution for a set of random variables, \( X_1, \ldots, X_n \) gives the probability of every combination of values (an \( n \)-dimensional array with \( v^n \) values if all variables are discrete with \( v \) values, all \( v^n \) values must sum to 1): \( P(X_1, \ldots, X_n) \)

<table>
<thead>
<tr>
<th></th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>circle</td>
<td>square</td>
<td>circle</td>
</tr>
<tr>
<td>red</td>
<td>0.20</td>
<td>0.02</td>
</tr>
<tr>
<td>blue</td>
<td>0.02</td>
<td>0.01</td>
</tr>
</tbody>
</table>

The probability of all possible conjunctions (assignments of values to some subset of variables) can be calculated by summing the appropriate subset of values from the joint distribution.

\[
P(red \wedge circle) = 0.20 + 0.05 = 0.25
\]

\[
P(red) = 0.20 + 0.02 + 0.05 + 0.3 = 0.57
\]

Therefore, all conditional probabilities can also be calculated.

\[
P(positive \mid red \wedge circle) = \frac{P(positive \wedge red \wedge circle)}{P(red \wedge circle)} = \frac{0.20}{0.25} = 0.80
\]
Probabilistic Classification

• Let \( Y \) be the random variable for the class which takes values \( \{y_1, y_2, \ldots, y_m\} \).

• Let \( X \) be the random variable describing an instance consisting of a vector of values for \( n \) features \( <X_1, X_2, \ldots, X_n> \), let \( x_k \) be a possible value for \( X \) and \( x_{ij} \) a possible value for \( X_i \).

• For classification, we need to compute \( P(Y=y_i \mid X=x_k) \) for \( i=1 \ldots m \).

• However, given no other assumptions, this requires a table giving the probability of each category for each possible instance in the instance space, which is impossible to accurately estimate from a reasonably-sized training set.

  – Assuming \( Y \) and all \( X_i \) are binary, we need \( 2^n \) entries to specify \( P(Y=\text{pos} \mid X=x_k) \) for each of the \( 2^n \) possible \( x_k \)’s since \( P(Y=\text{neg} \mid X=x_k) = 1 - P(Y=\text{pos} \mid X=x_k) \).

  – Compared to \( 2^{n+1} - 1 \) entries for the joint distribution \( P(Y, X_1, X_2, \ldots, X_n) \).
Bayes Theorem

\[ P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)} \]

Simple proof from definition of conditional probability:

\[ P(H \mid E) = \frac{P(H \land E)}{P(E)} \quad \text{(Def. cond. prob.)} \]

\[ P(E \mid H) = \frac{P(H \land E)}{P(H)} \quad \text{(Def. cond. prob.)} \]

\[ P(H \land E) = P(E \mid H)P(H) \]

QED: \[ P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)} \]
P(cancer) = 0.008 \hspace{1cm} P(!cancer) = 0.992
P(+ | cancer) = 0.98 \hspace{1cm} P(- | cancer) = 0.02
P(+ | !cancer) = 0.03 \hspace{1cm} P(- | !cancer) = 0.97

What do we say if the test returns positive?

MAP: Maximum *a posteriori* hypothesis