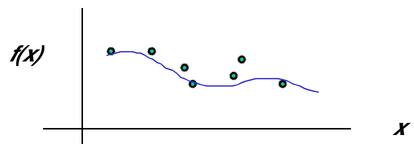
Generalization

The objective of learning is to achieve good *generalization* to new cases, otherwise just use a look-up table.

Generalization can be defined as a mathematical *interpolation* or *regression* over a set of training points:



Generalization

Weight Decay: an automated method of effective weight control

Adjust the bp error function to penalize the growth of unnecessary weights:

$$E = \frac{1}{2}\sum_j (t_j - o_j)^2 + \frac{\lambda}{2}\sum_i w_{ij}^2 \Rightarrow \Delta w_{ij} = \Delta w_{ij} - \lambda w_{ij}$$

where: λ = weight -cost parameter

w_{ij} is decayed by an amount proportional to its magnitude;

Generalization

A Probabilistic Guarantee

N = # hidden nodes m = # training cases

W = # weights ε = error tolerance (< 1/8)

Network will generalize with 95% confidence if:

1. Error on training set < $\varepsilon/2$
2. $m > O(\frac{W}{\varepsilon}\log_2\frac{N}{\varepsilon}) \approx m > \frac{W}{\varepsilon}$

Over-Training

Is the equivalent of over-fitting a set of data points to a curve which is too complex

Occam's Razor (1300s) : *"plurality should not be assumed without necessity"*

The simplest model which explains the majority of the data is usually the best

Preventing Over-training

- Use a separate *test* or *tuning set* of examples
- Monitor error on the test set as network trains
- Stop network training just prior to over-fit error occurring - *early stopping* or *tuning*
- Number of effective weights is reduced
- Most new systems have automated early stopping methods

Design & Training Issues

Design:

- Architecture of network
- Structure of artificial neurons
- Learning rules

Training:

- Ensuring optimum training
- Learning parameters
- Data preparation

Network Design

Architecture of the network: How many nodes?

Determines number of network weights

How many layers?

How many nodes per layer?

Input Layer	Hidden Layer	Output Layer
____________	____________	____________

Some Automated methods:

- augmentation (cascade correlation)
- weight pruning and elimination

Network Design

Architecture of the network: (Connectivity)

selective connectivity

shared weights

recursive connections

Network Design

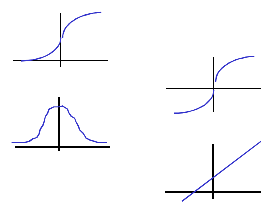
Structure of artificial neuron nodes

Choice of input integration:

- summed, squared and summed
- multiplied

Choice of activation (transfer) function:

- sigmoid (logistic)
- hyperbolic tangent
- Guassian
- linear
- soft-max



Network Design

Selecting a Learning Rule

- Generalized delta rule (steepest descent)
- Momentum descent
- Advanced weight space search techniques
- Global Error function can also vary

  - normal - quadratic - cubic

Network Training

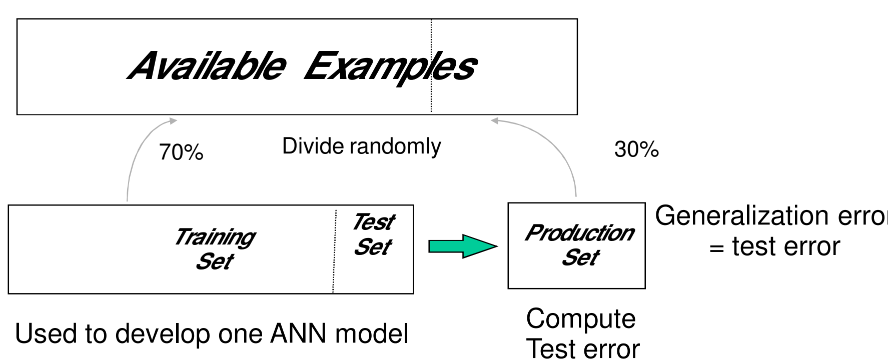
How do you ensure that a network has been well trained?

- ❖achieve good generalization accuracy on new examples
- ❖Establish a maximum acceptable error rate
- ❖Train the network using a *validation test set* to tune it
- ❖Validate the trained network against a separate test set which is usually referred to as a *test (production) set*

Network Training

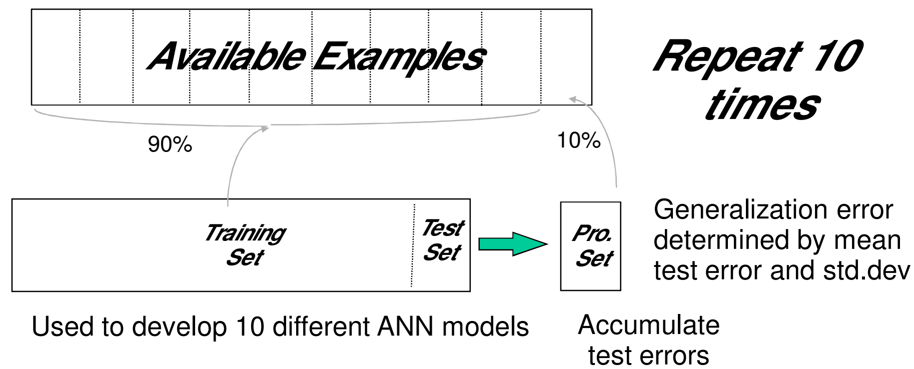
Large Sample

When the amount of available data is large ...



Network Training

Small Sample: Cross-validation



Network Training

How do you select between two ANN designs ?

A statistical test of hypothesis is required to ensure that a significant difference exists between the error rates of two ANN models

If Large Sample method has been used then apply McNemar's test

If Cross-validation then use a statistical *t* test for difference of two proportions

Network Training

Common ANN Parameters

		Typical	Range
learning rate -	η	0.1	0.01 - 0.99
momentum -	α	0.8	0.1 - 0.9
weight-cost -	λ	0.1	0.001 - 0.5

Fine tuning : adjust individual parameters at each node and/or connection weight

automatic adjustment during training

Network Training

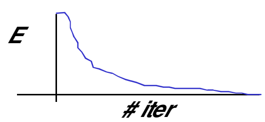
Network weight initialization

- Random initial values +/- some range
- Smaller weight values for nodes with many incoming connections
- Rule of thumb: initial weight range should be approximately $\pm \frac{1}{\# weights}$

  coming into a node

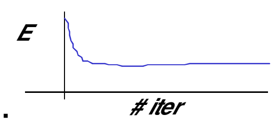
Typical Problems During Training

Optimal

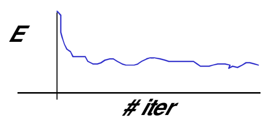


Steady, rapid decline in total error

But sometimes:



Seldom a local minimum - reduce learning or momentum parameter



Reduce learning parms.
- may indicate data is not learnable

Data Preparation

Garbage in → Garbage out

The quality of results relates directly to quality of the data

50%-70% of ANN development time will be spent on data preparation

The three steps of data preparation:

- Consolidation and Cleaning
- Selection and Preprocessing
- Transformation and Encoding

Data Preparation

Data Types and ANNs

Four basic data types- may be more:

- *nominal* discrete symbolic (blue,red,green)
- *ordinal* discrete ranking (1^{st}, 2^{nd}, 3^{rd})
- *interval* measurable numeric (-5, 3, 24)
- *continuous* numeric (0.23, -45.2, 500.43)

bp ANNs accept only continuous numeric values (typically 0 - 1 range)

Data Preparation

Consolidation and Cleaning

- Determine appropriate input attributes
- Consolidate (merge) data into working database
- Eliminate or estimate missing values
- Remove *outliers* (obvious exceptions)
- Determine prior probabilities of categories and deal with *volume bias*

Data Preparation

Selection and Preprocessing

- Select examples → random sampling

  Consider number of training examples?
- Reduce # of attributes

  remove redundant correlating attributes

  combine attributes (sum, multiply, difference)
- Reduce attribute value ranges

  group symbolic discrete values

  quantize continuous numeric values

Data Preparation

Transformation and Encoding

Nominal or Ordinal values

Transform to discrete numeric values

Encode the value 4 as follows:

one-of-N code (0 1 0 0 0) - five inputs

thermometer code (1 1 1 1 0) - five inputs

real value (0.4) - one input if ordinal

Consider relationship between values

(*single, married*) *vs.* (*youth, adult*)

Data Preparation

Transformation and Encoding

Interval or continuous numeric values

normalization of values:

- Euclidean: *n = x/sqrt(sum of all x^2)*
- Percentage: *n = x/(sum of all x)*
- Variance based: *n = (x - (mean of all x))/variance*

Scale values using a linear transform if data is uniformly distributed or use non-linear (log, power) if skewed distribution

Data Preparation

Transformation and Encoding

Interval or continuous numeric values

Encode the value 1.6 as:

- Single real-valued number (0.16) - OK!
- Bits of a binary number (010000) - BAD!
- one-of-N quantized intervals (0 1 0 0 0)
  - NOT GREAT! - discontinuities
- distributed (fuzzy) overlapping intervals
  (0.3 0.8 0.1 0.0 0.0) - BEST!

After-Training Analysis

Examining the neural net model:

- Visualizing the constructed model
- Detailed network analysis

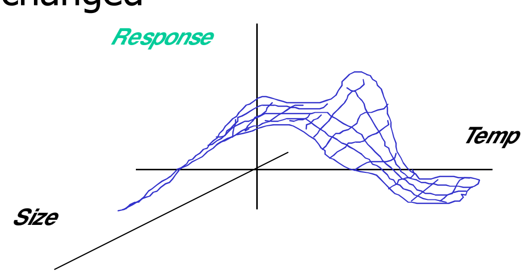
Sensitivity analysis of input attributes:

- Analytical techniques
- Attribute elimination

After-Training Analysis

Visualizing the Constructed Model

Graphical tools can be used to display output response as selected input variables are changed



After-Training Analysis

Detailed network analysis

- Hidden nodes form internal representation
- Manual analysis of weight values often difficult - graphics very helpful
- Conversion to equation, executable code
- Automated ANN (computational models) to symbolic logic conversion is a hot (new/ hard) area of research

After-Training Analysis

Sensitivity analysis of input attributes

- Analytical techniques
  - factor analysis

  network weight analysis
- Feature (attribute) elimination
  - forward feature elimination
  - backward feature elimination

The ANN Application Development Process

Guidelines for using neural networks

1. Try the best existing method first
2. Get a **big** training set
3. Try a NN without hidden units
4. Use a sensible coding for input variables
5. Consider methods of constraining network
6. Use a test set to prevent over-training
7. Determine confidence in generalization through cross-validation

Applications

- Pattern Recognition *(reading zip codes)*
- Signal Filtering *(reduction of radio noise)*
- Data Segmentation *(detection of seismic onsets)*
- Data Compression *(TV image transmission)*
- Database Mining *(marketing, finance analysis)*
- Adaptive Control *(vehicle guidance)*
- Handwriting recognition
- Face recognition
- Optical character recognition (OCR)

Applications

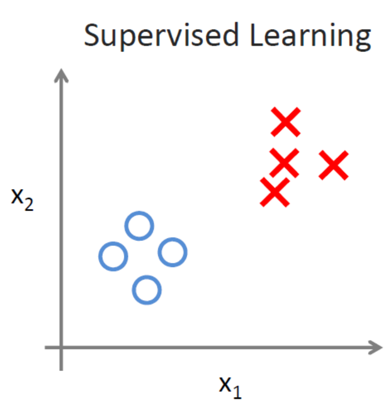
- Text to Speech (NetTalk)
- Fraud detection

  9 of top 10 US credit card companies use Falcon uses neural nets to model customer behavior, identify fraud claims
- Prediction & Financial Analysis

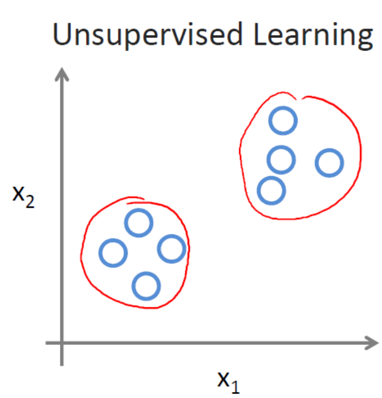
  In Banks: financial forecasting, investing, marketing analysis
- control & optimization
  - Intel – computer chip manufacturing quality control
  - AT&T (cell phones) – echo & noise control in phone lines (filters and compensates)
  - Ford engines utilize neural net chip to diagnose misfirings, reduce emissions

Learning

Classification

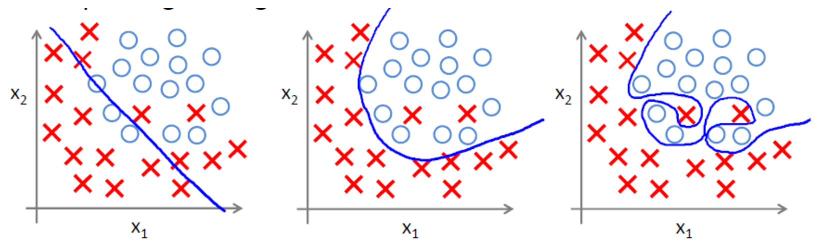


Clustering



Data Models

Fitting the data with a model



Under fitting Fitting Over Fitting

Over fitting model

Over fittintg: If you have too many parameters (features) , your model may be learned to fit the training set very well, with almost 0 error. But fail to generalize to new examples.

Preventing over fitting

More ways:

1- Reduce the number of features (attributes)

- Manually, check which feature to keep
- Model selection (Rough Set –Attribute reduction algorithms)

2- Regularization

- Keep all features but reduce their effect (different methods)

Rough Set Theory

- Can be used for feature selection, feature extraction, data reduction, and others
- Identifies partial or total dependencies in data, eliminates redundant data
- Simple Idea:

  If the number of equivalent types(element sets) derived from attribute set A is the same as that derived from A – ai, then attribute ai is regarded as redundant.

Rough Set Theory

- The reduct: the set of attributes after removing unwanted ones
- heuristic search is used, Find the best reduct as search problem
- Common Algorithms of Attribute Reduct
  - Discrenibility Matrix
  - Quick Reduct
  - Entropy-Based algorithm