Master of Technology

Unit 2/6: Computational Intelligence I

Modelling and Problem Solving with NN

Dr. Zhu Fangming
Institute of Systems Science
National University of Singapore

Objectives

- To introduce the modelling and problem solving approaches with NN
- To discuss some important issues on using NN for classification problems

Outline

- Problem analysis and insights
- Model selection and network building
- NN for classification – concepts and issues
- Principal Component Analysis & NN

When to Use NN

- **Viewing types of applications**
  - » **NN will be suitable for applications that involve certain kind of *pattern recognition / classification, pattern clustering / grouping and forecasting time series***
  - » **Possible applications:**
    pattern recognition; optimisation; forecasting; general mapping / functional mapping; fault diagnosis; data compression; control / decision making; associative memory; clustering / grouping; …
- **Viewing availability of domain knowledge / data**
  - » **knowledge can be summarised as rules**
    - ♦ **complete and certain: Rule-Based System**
    - ♦ **incomplete or uncertain: Fuzzy System (rule-based)**
  - » **knowledge appears as successful cases**
    - ♦ **Case-Based Reasoning**
  - » **knowledge / historical data appears as patterns**
    - ♦ **NN system**

How NN Can be Applied in Intelligent Systems

- **NN system**
  - » **to build an NN to solve an entire application**

- **Hybrid system**
  - » **to build an NN module which will be used with other module(s) to form an intelligent system to solve an application**
  - » **to serve as a function or simulation that will help other module of the intelligent system**

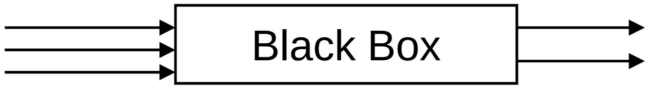
- **Rule extraction**
  - » **to “learn” or “discover” rules from the given data, and the rules will be used by other intelligent module of the system**

Developing an NN System

- A general development methodology:
  - » Problem formulation
    - ♦ Identify the application / problem type
      - • Association, Classification, Diagnosis, Control /decision making, Data compression/ fusion, Forecasting /prediction, Function approximation /general mapping, Grouping/clustering, Optimization, Pattern recognition, …
  - » System modelling / design
    - ♦ Identify all input, output and related variables
    - ♦ NN design
  - » Development / implementation
    - ♦ Data preparation, training, testing
  - » Evaluation
    - ♦ Refining, optimization
  - » Operation / Validation / deployment/ Maintenance
- Key tasks:
  - » Identify objectives and variables
  - » Check data availability
  - » Choose a suitable NN architecture

Modelling and Design

- **NN models can be general**



- **System variables need to be identified and training data collected.**

- **Important factors for modelling:**
  - » **Relevant variables, their types & ranges**
  - » **Preparation of appropriate training, testing and validation data sets**
  - » **Boundary conditions (out of range or illegal values)**
  - » **Accuracy needed to realize useful solutions**
  - » **Balance between accuracy and complexity**

Implementation
— Building NN & Pre-processing Data

- **Network architecture:**
  - » **Initially dictated by the problem/application type**
  - » **Determined by the system dynamics and experimentation, when more than one net architectures can solve the problem**
- **Training/test data set**
  - » **Perform statistical analyses to support data set choices**
  - » **Select representative training set**
  - » **Divide training set & testing set appropriately**
- **Pre-processing the data**
  - » **It is an important part of the overall modelling/solution process. It is estimated that this phase requires more than 50% of the total solution effort.**
  - » **Use basic analysis tools for both single and multivariate analyses methods: analyze, plot, clean & transform the data. When appropriate, also perform other analyses: e.g. Cluster analysis, Principal component analysis (PCA), Correlation analyses, Discriminant analysis, Factor analysis**

Implementation — Data Coding

- **Linearly independent codes, e.g.:**

	Code 1	**Code 2**
Input No.	**1 2 3 4 5**	**5 4 3 2 1**
	1 0 0 0 0	**1 1 1 1 1**
	0 1 0 0 0	**1 1 1 1 0**
	0 0 1 0 0	**1 1 1 0 0**
	0 0 0 1 0	**1 1 0 0 0**
	0 0 0 0 1	**1 0 0 0 0**

- **Ordinal (non interval) data such as risk level (high, medium, low)**
  - » **Can be coded numerically as 3, 2, 1 (one variable)**
  - » **Binary codes can be used (e.g. above) (multiple variables)**

- **For non-ordinal items (that is nominal or qualitative variables, e.g. marital status)**
  - » **Can be coded numerically:**

    e.g.: single=1, married=2, divorced=3, widowed=4

Implementation
— Transforms and Indicators

- ***Log:*** **Take the natural or decimal logarithm of each numeric data value x, y=log(x). This is useful when the range of a variable is large.**

- ***Delta:*** **Compute the difference between the current value and previous value $\Delta x_i = x_i - x_{i-1}$. This removes the base or minimum value a variable has and is used to remove trends from non-stationary data.**

- ***Log-Delta:*** **Compute the difference between the logarithm of the current and logarithm of the previous values $y = \log(x_i) - \log(x_{i-1})$.**

- ***Moving average:*** **The k-term moving average is the average of the moving window of k values of the x_i data, i.e.**

$$MA_k = \frac{1}{k}\sum_{n=0}^{k-1} x_{i-n}$$

Implementation
— Transforms and Indicators (cont.)

- *Exponential average:* **The weighted average of a sequence of variant values (the most recent points are given the most weight since 0 < α < 1)**

$$\mathrm{EA}(t) = (1-\alpha)\sum_{j=0}^{m} \alpha^{j} \mathbf{x}_{t-j}$$

- ***Normalization*: For varying data with large values, normalizion to values y ∈ [0, 1] using the formula**

$$y = \frac{x - \min(x)}{\max(x) - \min(x)}$$

can be more *meaningful* and easier to process for NN inputs.

- ***Normalized z-scores*: transforms data to have zero mean and unit standard deviations**

$$z = \frac{x - \mu}{\sigma}$$

- **Other transforms for special applications:**
  - » **Wavelet transform, Fourier transform, etc.**

ISS National University of Singapore

Implementation
— Testing / Evaluation

- **Testing the Generalization ability of a trained NN**
  - » **Look for good performance on a validation set and test set**
  - » **Changing the training algorithm**
- **The performance varies with training/ solution procedures**

- **Network optimization should be performed after training/testing (eliminate redundant unneeded nodes and the corresponding weights – is called *‘pruning’*)**

- **Periodic performance testing is essential to verify model’s accuracy - environmental changes can cause the data to change thereby afflicting the performance of the developed model**

NN for Classification: Further Discussion

NN for classification

- **Classification is one of the most active research and application areas of NN**

- **An NN for a classification problem can be viewed as a mapping function from n-dimensional input space to a *M*-vectored output as the classification decision**

- **NNs are universal approximators and in theory can approximate any nonlinear function closely. However, the mapping function presented by a network is not perfect due to the local minima problem, sub-optimal network architecture and finite sample data in NN training.**

NN for Classification: Further discussion — Learning & generalization

Learning

- **The ability to approximate the underlying behavior adaptively from the training data.**

Generalization

- **The ability to predict well beyond the training data.**
- **It is a more desirable and critical feature because the most common use of a classifier is to make good prediction on new or unknown objects.**
- **Powerful data fitting or function approximation capability of NN also makes them susceptible to the overfitting problem.**
- **The symptom of an overfitting model:**
  - » **Fits the training sample very well but has poor generalization capability when used for prediction purpose.**

National University of Singapore

NN for Classification: Further discussion — Reducing prediction error

- **NN often tends to fit the training data very well, but the potential risk is the over-fitting that causes high variance in generalization.**
- **NN classifiers belong to unstable prediction methods**
  - » **small changes in the training sample could cause large variations in the test results**
- **Research efforts have been devoted to developing methods to reduce the prediction error**
- **Ensemble method or combining multiple classifiers is another research area to reduce generalization error**

NN for Classification: Further discussion — Combining multiple classifiers

- An ensemble can be formed by
  - » multiple network architectures
  - » same architecture trained with different algorithms
  - » NNs using different initial random weights
  - » NNs developed by training with different data
  - » mixed combination of NNs with classifiers using other techniques
- Combining multiple classifiers
  - » voting schemes
  - » via simple average of outputs from individual classifiers
  - » weighted averaging that treats the contribution or accuracy of component classifiers differently
  - » ranking based information

NN for Classification: Further discussion — Combining multiple classifiers …

- **The ensemble method works better if different classifiers in the ensemble disagree with one another strongly. However, the averaging effect may decrease the sensitivity of the classifier to the new data.**

- **Ensemble methods are very effective when individual classifiers are uncorrelated. To reduce correlation among component classifiers:**
  - » **Classifiers based on different features**
  - » **Training with different data sets**

- **Issues remain, have not completely solved**
  - » **Choice of classifiers**
  - » **Size of ensemble**
  - » **Optimal way to combine individual classifiers**
  - » **Under what conditions combining is most effective and what methods should be included**

ISS National University of Singapore

NN for Classification: Further discussion —Feature selection & NN construction

- **Selection of a set of appropriate input features *-feature selection-* is an important issue in building NN and other classifiers.**
  - » **It is often necessary and beneficial to limit the number of input features in a classifier in order to have a good predictive and less computational intensive model**
  - » **It is extremely important step in *Data Mining***

- **Research efforts are increasing in developing feature selection or dimension reduction approaches for neural network classifiers. However, most of the methods are *heuristic* in nature.**

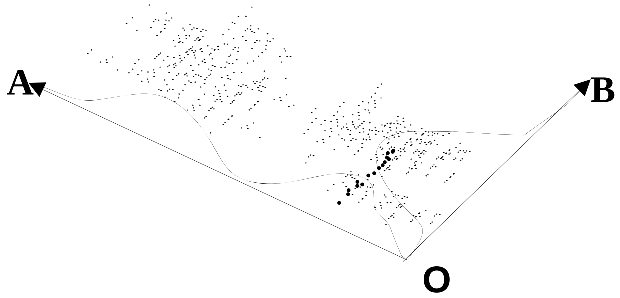
- **PCA is one of the most popular methods in feature selection**
  - » **Often used as a pre-processing method in NN training**
  - » **Problems with PCA**
    - ♦ **It is an unsupervised learning procedure**
    - ♦ **It does not reflect on the input features directly**
    - ♦ **It is a linear dimension reduction technique**

Analysis Approaches
— Principal Component Analysis (PCA) —

- PCA is an extremely popular method from multivariate statistics for achieving ***data compression and dimensionality reduction***

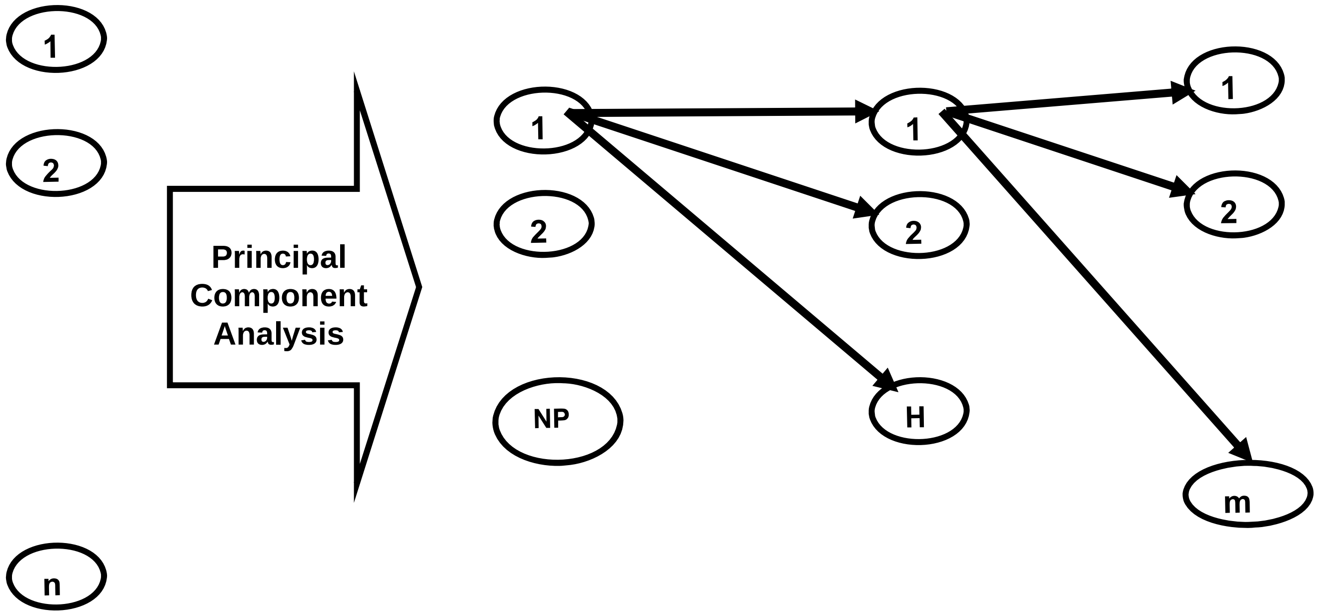
- The aim is to find a set of *m* orthogonal vectors in the data space that account for as much as possible of the variance in the original data and then project the data from their original *n*-dimensional space onto the *m*-dimensional subspace spanned by these vectors, where $m < n$.

- E.g.
  - » OA is the first principal component direction of the distribution that generates the cloud of points
  - » OB is the second principal component direction



Principal Component Analysis application in building Hybrid Neural Networks

- The original input variables are subjected to PCA and then only a selected few PCs are input to the hidden layer in a multi-layer perceptron. The final network so formed is trained by BP algorithm



Applications of Hybrid Principal Component Analysis Neural Networks

- **Human face recognition (Abidi et al, 1995)**
- **Prediction of software faults (Khoshgoftaar and Szabo, 1996)**
- **Identification of calcium channel blockers that cause heart attack (Viswanadhan et al, 1996)**
- **Operation of a fed-batch bioreactor (Kurtanjek et al, 1997)**
- **Prediction in meteorology and oceanography (Hsieh and Tang, 1998)**
- **Long range forecasting of Indian monsoon (Guhatakurta, 1999)**
- **In bioinformatics, classification and prediction of cancers using gene expression profiles (Javed Khan et al, 2001)**
- **Identification of microcalcification clusters in digital mammograms (Papadopoulos, 2002)**

ISS National University of Singapore

NN for Classification: Further discussion —Feature selection & NN construction

- **Some other work on feature selection and NN construction**
  - » **Starts with the whole set of available feature variables and then for each attribute variable, the accuracy of the network is evaluated with all the weights associated with that variable set to zero. The variable that gives the lowest decrease in accuracy is removed (Setiono & Liu)**
  - » **Many techniques of weight elimination and node pruning are often used to remove unnecessary linking weights or input nodes during the network training.**

NN for Classification: Further discussion — Misclassification costs

- **The assumption of equal cost consequences of misclassification**
  - ♦ **simplifies the model development**
  - ♦ **but does not represent many real problems well**

- **Applications where uneven misclassification costs are more appropriate:** ***quality assurance, bankruptcy prediction, credit risk analysis, medical diagnosis, fraudulent transactions***

- **Under the assumption of equal consequences of misclassification, a classifier tends to bias toward the larger groups that have more observations in the training sample.**
- **Some of possible ways to help**
  - » **The appropriate use of cost information**
    - ♦ **E.g.: using weighted error function to incorporate the prior knowledge about the relative class importance or different misclassification costs.**
  - » **Using equal number of examples from each group for training**

NN for Classification: Further discussion — Conclusion

- **Many issues still remain unsolved or incompletely solved**
  - » **Feature variable selection**
  - » **Classifier combination**
  - » **Uneven misclassification treatment**
  - » **Model design and selection**
  - » **Sample size issues**

Conclusions

- **Neural networks can often be used to create an accurate model of a system or a process if good training and test data are available.**

- **NNs can learn arbitrary mappings from training data, but the degree of success depends on**
  - » **the quality of the training data,**
  - » **its pre-processing**
  - » **the network architecture (type, structure and parameters)**
  - » **the training algorithm**

- **An understanding of the underlying application process is important as this can help us make right choices of network type, network parameters and pre-processing of data.**

- **Data analysis, mathematics and statistical tools are indispensable aids in problem formation, design and implementation of neural network models.**