Part 5: CUTTING EDGE DECISION SUPPORT TECHNOLOGIES

- Neural Computing
- Genetic Algorithms
- Fuzzy Logic

- Integration

Chapter 17: Neural Computing: The Basics

- **Artificial Neural Networks (ANN)**
- **Mimics How Our Brains Work**
- **Machine Learning**

17.1 Opening Vignette: Maximizing the Value of the John Deere & Co. Pension Fund

The Problem

- Managing the Pension Fund of John Deere & Co.
- About $1 Billion Managed Internally by Corporate Finance
- Want to Achieve a Better Return on Investment

The Solution

- Neural Computing (Since 1993)

The Process

- Build an individual, ANN for each of the 1000 largest U.S. Corporations
- Historical Data: 40 inputs
- Weekly, data updates into a stock prediction model (Figure 17.1)
- Ranks the stocks based on *anticipated performance*
- Selects a portfolio of the 100 top stocks

Results

- **Returns are well ahead of industry benchmarks**

17.2 Machine Learning: An Overview

- ANN to automate complex decision making
- Neural networks learn from past experience and improve their performance levels
- Machine learning: Methods that teach machines to solve problems, or to support problem solving, by applying historical cases

Complications

- Many models of learning
- Match the learning model with problem type

What is Learning?

- Accomplished by analogy, discovery and special procedures; by observing; or by analyzing examples
- Can improve the performance of AI methods
- Is a Support area of AI

Learning as Related to AI

- Learning systems demonstrate *interesting* learning behaviors
- No claims about learning as well as humans or in the same way
- Learning systems are not defined very formally; Implications are not well understood
- Learning in AI involves the manipulation of symbols (not numeric information)

Examples of Machine Learning Methods

- Neural Computing
- Inductive Learning
- Case-based Reasoning and Analogical Reasoning
- Genetic Algorithms
- Statistical methods
- Explanation-based Learning

17.3 An Overview of Neural Computing

- **Constructing computers that mimic certain processing capabilities of the human brain**
- **Knowledge representations based on**
  - **Massive parallel processing**
  - **Fast retrieval of large amounts of information**
  - **The ability to recognize patterns based on historical cases**

Neural Computing = Artificial Neural Networks (ANNs)

17.4 The Biology Analogy

Biological Neural Networks

- Neurons: Brain Cells
  - Nucleus (at the center)
  - Dendrites provide inputs
  - Axons send outputs
- Synapses increase or decrease connection strength and causes excitation or inhibition of subsequent neurons

Artificial Neural Networks (ANN)

- A model that emulates a biological neural network
- Software simulations of the massively parallel processes that involve processing elements interconnected in a network architecture
- Originally proposed as a model of the human brain’s activities
- The human brain is ***much more*** complex

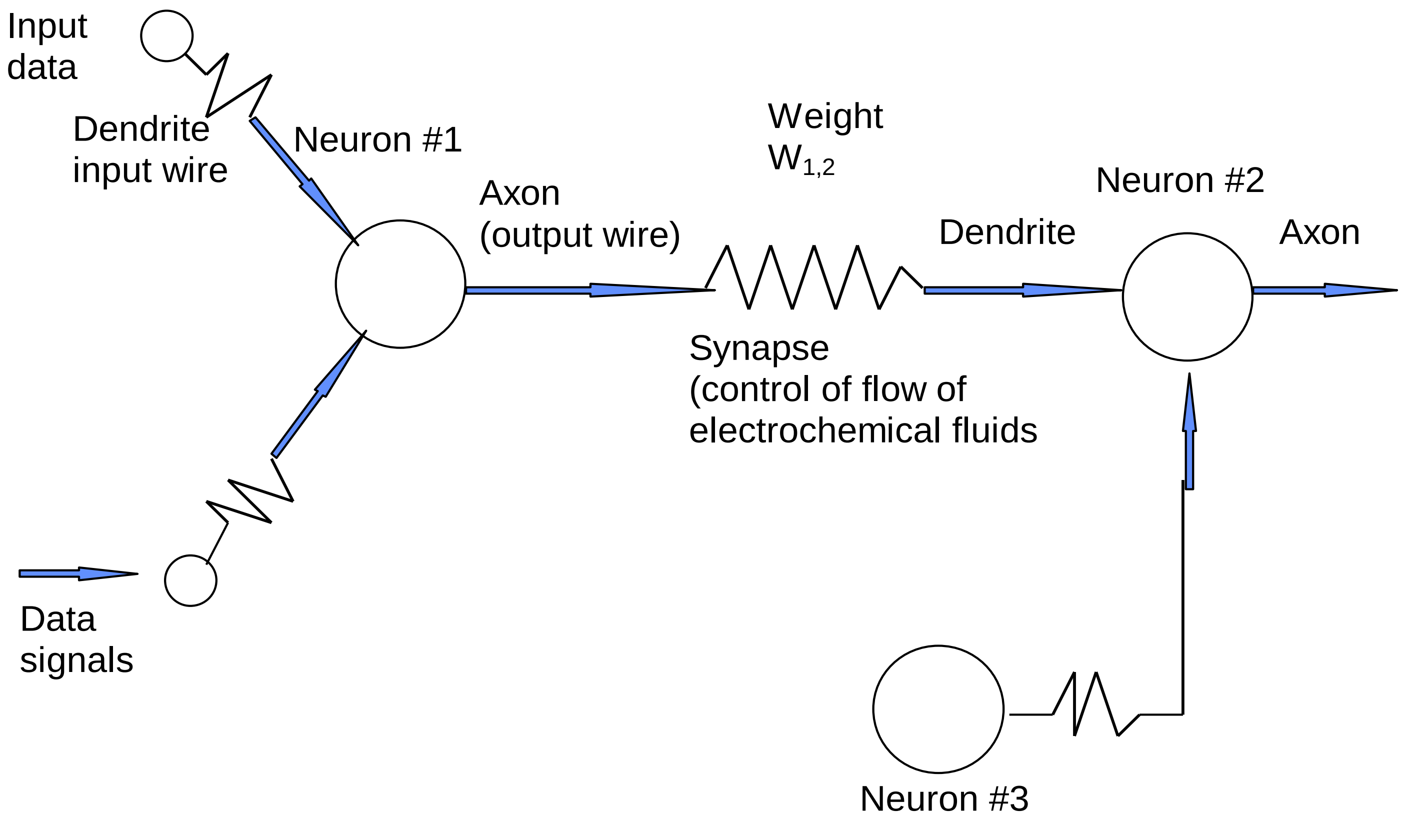

FIGURE 17.3 Three Interconnected Artificial Neurons

AIS In Focus 17.1:

The Relationship Between Biological and Artificial Neural Networks

Biological	Artificial
Soma	Node
Dendrites	Input
Axon	Output
Synapse	Weight
Slow Speed	Fast Speed
Many Neurons (10^9)	Few Neurons (dozen to hundreds of thousands)

Medsker and Liebowitz [1994, p. 163]

17.5 Neural Network Fundamentals

- **Components and Structure**
  - Processing Elements
  - Network
  - Structure of the Network
- **Processing Information in the Network**
  - Inputs
  - Outputs
  - Weights
  - Summation Function

jth Neuron (Figure 17.6)

- **Transformation (Transfer) Function**
- **Sigmoid Function (Logical Activation Function)**

$$Y_T = \frac{1}{1 + e^{-Y}}$$

  - **where Y_T is the transformed (normalized) value of Y**

Learning

Three Tasks

1. Compute Outputs

2. Compare Outputs with Desired Targets

3. Adjust Weights and Repeat the Process

- **Set the weights by either some rules or randomly**
- **Set Delta = Error = actual output minus desired output for a given set of inputs**
- **Objective is to *Minimize* the Delta (Error)**
- ***Change* the weights to reduce the Delta**

- **Information processing: pattern recognition**
- **Different learning algorithms**

AIS In Focus 17.4: How Patterns Are Presented and Recognized

Desired Outputs

Pattern										Class
0	0	0	0	0	0	0	0	0	0	down
1	1	1	1	0	0	0	0	0	0	left
1	1	1	0	0	0	0	1	1	1	valley
0	1	0	1	0	1	0	1	0	1	alternating
0	0	0	0	0	0	1	1	1	1	right
0	0	0	1	1	1	1	0	0	0	hill
1	1	1	1	1	1	1	1	1	1	up

Historical Cases

Pattern											Historical Interpretation
Test cases											
#1	1	1	0	0	1	1	0	0	1	1	alternating
#2	0	0	0	0	0	1	0	0	0	1	right
#3	1	1	0	1	1	1	0	0	0	0	left
#4	0	0	1	1	0	1	0	1	0	1	alternating
#5	0	0	0	1	1	0	1	1	0	0	alternating
Test cases											
#6	1	1	1	0	1	1	1	0	1	1	alternating
#7	0	0	1	0	0	0	1	1	1	1	right
#8	0	1	0	1	1	0	1	1	0	0	alternating
#9	1	1	1	0	0	0	0	0	0	0	left
#10	1	1	1	1	0	0	1	1	1	1	valley

Source: C. W. Engel and M. Cran, " Pattern Classifications: A Neural Network Competes with Humans," *PC AI,* May/June 1991. Used with permission.)

17.6 Neural Network Applications Development

- Preliminary steps of system development are done
- ANN Application Development Process
  - Collect Data
  - Separate into Training and Test Sets
  - Define a Network Structure
  - Select a Learning Algorithm
  - Set Parameters, values, Initialize Weights
  - Transform Data to Network Inputs
  - Start Training, and Determine and Revise Weights
  - Stop and Test
  - Implementation: Use the Network with New Cases

17.7 Data Collection and Preparation

- Collect data and separate into a ***training set*** and a ***test set***
- Use ***training cases*** to adjust the weights
- Use ***test cases*** for network validation

17.8 Neural Network Architectures

Representative Architectures

- Associative Memory Systems
  - Associative memory - ability to recall complete situations from partial information
  - Systems correlate input data with stored information
  - Hidden Layer
  - Three, Sometimes Four or Five Layers

Recurrent Structure

- **Recurrent network (double layer) - each activity goes through the network more than once before the output response is produced**

- **Uses a *feedforward* and *feedbackward* approach to adjust parameters to establish arbitrary numbers of categories**

- **Example: Hopfield**

17.9 Neural Network Preparation

(Non-numerical Input Data (text, pictures): preparation may involve simplification or decomposition)

- Choose the learning algorithm
- Determine several parameters
  - Learning rate (high or low)
  - Threshold value for the form of the output
  - Initial weight values
  - Other parameters
- Choose the network's structure (nodes and layers)
- Select initial conditions
- Transform training and test data to the required format

17.10 Training the Network

- Present the ***training data*** set to the network
- ***Adjust weights*** to produce the desired output for each of the inputs
  - Several iterations of the complete training set to get a consistent set of weights that works for all the training data

17.11 Learning Algorithms

- **Two Major Categories Based On Input Format**
  - **Binary-valued (0s and 1s)**
  - **Continuous-valued**
- **Two Basic Learning Categories**
  - **Supervised Learning**
  - **Unsupervised Learning**

Supervised Learning

- For a set of inputs with known (desired) outputs
- Examples
  - Backpropagation
  - Hopfield network

Unsupervised Learning

- Only input stimuli shown to the network
- Network is self-organizing
- Number of categories into which the network classifies the inputs can be controlled by varying certain parameters
- Examples
  - Adaptive Resonance Theory (ART)
  - Kohonen Self-organizing Feature Maps

How a Network Learns

Single neuron - learning the inclusive OR operation

Two input elements, X_1 and X_2

	Inputs		
Case	X_1	X_2	Desired Results
1	0	0	0
2	0	1	1 (positive)
3	1	0	1 (positive)
4	1	1	1 (positive)

Table 17.1 in Excel

TABLE 17.1 Example of Supervised Learning

Step	X_1	X_2	Z	Initial W_1	Initial W_2	Y	Delta	Final W_1	Final W_2
1	0	0	0	0.1	0.3	0	0.0	0.1	0.3
	0	1	1	0.1	0.3	0	1.0	0.1	0.3
	1	0	1	0.1	0.5	0	1.0	0.1	0.5
	1	1	1	0.3	0.5	1	0.0	0.3	0.5
2	0	0	0	0.3	0.5	0	0.0	0.3	0.5
	0	1	1	0.3	0.5	0	0.0	0.3	0.5
	1	0	1	0.3	0.7	0	1.0	0.5	0.7
	1	1	1	0.5	0.7	1	0.0	0.5	0.7
3	0	0	0	0.5	0.7	0	0.0	0.5	0.7
	0	1	1	0.5	0.7	1	0.0	0.5	0.7
	1	0	1	0.5	0.7	0	1.0	0.7	0.7
	1	1	1	0.7	0.7	1	0.0	0.7	0.7
4	0	0	0	0.7	0.7	0	0.0	0.7	0.7
	0	1	1	0.7	0.7	1	0.0	0.7	0.7
	1	0	1	0.7	0.7	1	0.0	0.7	0.7
	1	1	1	0.7	0.7	1	0.0	0.7	0.7

Parameters: alpha = 0.2; threshold = 0.5.

- A step function evaluates the summation of input values

- **Calculating outputs**
  - Measure the error (delta) between outputs and desired values
  - Update weights, reinforcing correct results

At any step in the process for a neuron, *j*, we get

$$\text{Delta} = Z_j - Y_j$$

where *Z* and *Y* are the desired and actual outputs, respectively

The updated weights are

$$W_i\,(\text{final}) = W_i\,(\text{initial}) + \text{alpha} \times \text{delta} \times X_1$$

where alpha is the learning rate parameter

- **Weights initially random**
- **The *learning rate* parameter, alpha, is set low**
- **Delta is used to derive the final weights, which then become the initial weights in the next iteration (row)**
- **Threshold value parameter: sets Y to 1 in the next row if the weighted sum of inputs is greater than 0.5; otherwise, to 0**

17.12 Backpropagation

- Backpropagation (back-error propagation)
- Most widely used learning
- Relatively easy to implement
- Requires training data for conditioning the network before using it for processing other data
- Network includes one or more hidden layers
- Network is considered a *feedforward* approach

Continue

- **Externally provided correct patterns are compared with the neural network output during training (supervised training)**
- **Feedback is used to adjust the weights until all training patterns are correctly categorized**

- **Error is backpropogated through network layers**
- **Some error is attributed to each layer**
- **Weights are adjusted**

- **A large network can take a very long time to train**
- **May not converge**

17.13 Testing

- **Test the network after training**
- **Examine network performance: measure the network's classification ability**
- **Black box testing**
- **Do the inputs produce the appropriate outputs?**

Continue

- **Not necessarily 100% accurate**
- **But may be better than human decision makers**
- **Test plan should include**
  - **Routine cases**
  - **Potentially problematic situations**
- **May have to retrain**

17.14 Implementation

- **Frequently requires proper interfaces with other CBIS and user training**
- **Gain confidence of the users and management early**

Neural Computing Paradigms

Decisions the builder must make

- Size of training and test data
- Learning algorithms
- Topology: number of processing elements and their configurations
- Transformation (transfer) function
- Learning rate for each layer
- Diagnostic and validation tools

Results in the Network's Paradigm

17.15 Programming Neural Networks

- An ANN can be programmed with
  - A programming language
  - A programming tool
  - Both
- Tools (shells) incorporate
  - Training algorithms
  - Transfer and summation functions
- May still need to
  - Program the layout of the database
  - Partition the data (test data, training data)
  - Transfer the data to files suitable for input to an ANN tool

Development Tool Examples

- NNetSheet (spreadsheet-based)
- Neuralyst for Excel (front-ended by a spreadsheet)
- KnowledgeNet and NeuroSMARTS (work with expert systems)
- NeuralWorks, Explorer and Brainmaker (stand alone environments)
- Languages (C++)

17.16 Neural Network Hardware

- **Advantages of massive parallel processing**
- **Greatly enhances performance**

Possible Hardware Systems for ANN Training

- Faster general purpose computers
- General purpose parallel processors
- Neural chips
- Acceleration boards
- Pentium MMX and II (Graphics Processors) (1998)

17.17 Benefits of Neural Networks

- Usefulness for pattern recognition, learning, classification, generalization and abstraction, and the interpretation of incomplete and noisy inputs
- Specifically - character, speech and visual recognition
- Potential to provide some of human problem solving characteristics
- Ability to tackle new kinds of problems
- Robustness
- Fast processing speed
- Flexibility and ease of maintenance
- Powerful hybrid systems

17.18 Limitations of Neural Networks

- Do not do well at tasks that are not done well by people
- Lack explanation capabilities
- Limitations and expense of hardware technology restrict most applications to software simulations
- Training times can be excessive and tedious
- Usually requires large amounts of training and test data

17.19 Neural Networks and Expert Systems

ANN and ES technologies are so different that they can *complement* each other

- **Expert systems represent a logical, symbolic approach**
- **Neural networks are model-based and use numeric and associative processing**
- **Main features of each (Table 17.2)**

TABLE 17.2 Comparing Expert Systems and Artificial Neural Networks.

	Expert Systems	Artificial Neural Networks
1	Have user development facilities, but systems should preferably be developed by skilled developers because of knowledge acquisition complexities.	Have user development facilities and can be easily developed by users with a little training.
2	Takes a longer time to develop. Experts must be available and willing to articulate the problem-solving process.	Can be developed in a short time. Experts only need to identify the inputs, outputs and a wide range of samples.
3	Rules must be clearly identified. Difficult to develop for intuitively-made decisions.	Does not need rules to be identified. Well suited for decisions made intuitively.
4	Weak in pattern recognition and data analysis such as forecasting.	Well suited for such applications but need wide range of sample data.
5	Not fault tolerant.	Highly fault tolerant.
6	Changes in problem environment warrant maintenance.	Highly adaptable to changing problem environment.
7	The application must fit into one of the knowledge representation schemes (explicit form of knowledge).	ANNs can be tried if the application does not fit into one of ES's representation schemes.
8	The performance of the human expert who helped create the ES places a theoretical performance limit for the ES.	ANNs may outperform human experts in certain applications such as forecasting.

9	Have explanation systems to explain why and how the decision was reached. Required when the decision needs explanation to inspire confidence in users. Recommended when the problem-solving process is clearly known.	Have no explanation system and act like black boxes.
10	Useful when a series of decisions is made in the form of a decision tree and when, in such cases, user interaction is required.	Useful for one-shot decisions.
11	Useful when high-level human functions such as reasoning and deduction need to be emulated.	Useful when low-level human functions such as pattern recognition need to be emulated.
12	Are not useful to validate the correctness of ANN system development.	In certain cases are useful to validate the correctness of ES development.
13	Use a symbolic approach (people-oriented).	Use a numeric approach (data-oriented).
14	Use logical (deductive) reasoning.	Use associative (inductive) reasoning.
15	Use sequential processing.	Use parallel processing.
16	Are closed.	Are self-organizing.
17	Are driven by knowledge.	Are driven by data.
18	Learning takes place outside the system.	Learning takes place within the system.
19	Are built through knowledge extraction.	Are built through training, using examples.

Source: Items 1-12 are from J. R. Slater et al. " On Selecting Appropriate Technology for Knowledge Systems," *Journal of Systems Management*, October 1993, p. 15.

Expert Systems

- Especially good for closed-system applications (literal and precise inputs, logical outputs)
- Reason by using established facts and pre-established rules

Major Limitations

- Experts do not always think in terms of rules
- Experts may not be able to explain their line of reasoning
- Experts may explain it incorrectly
- Sometimes difficult or impossible to build the knowledge base

Neural Computing Use

Neural Networks in Knowledge Acquisition

- **Fast identification of implicit knowledge by automatically analyzing cases of historical data**
- **ANN identifies *patterns* and relationships that may lead to rules for expert systems**
- **A trained neural network can rapidly process information to produce associated facts and consequences**

17.20 Neural Networks For Decision Support

- Inductive means for gathering, storing and using experiential knowledge
- Neural network-based DSS to appraise real estate in New York (90% accurate)
- Forecasting
- ANN in decision support: Easy sensitivity analysis and partial analysis of input factors

Continue

- **The relationship between a combined expert system, ANN and a DSS (Figure 17.13)**
- **ANN can expand the boundaries of DSS**

Summary

- ANN conceptually similar to biological systems
- Human brain composed of billions of neuron cells, grouped in interconnected clusters
- Neural systems are composed of interconnected processing elements called artificial neurons that form an artificial neural network
- Can organize an ANN many different ways

- **Weights express the relative strength (or importance) given to input data**
- **Each neuron has an activation value**
- **An activation value is translated to an output \through a transformation (transfer) function**
- **Artificial neural networks learn from historical cases**
- **Learning is done with algorithms**

Continue

- **Supervised learning**
- **Testing is done by using historical data**
- **Neural computing is frequently integrated with traditional CBIS and ES**
- **Many ANNs are built with tools that include the learning algorithm(s) and other computational procedures**
- **ANNs lend themselves to parallel processing. However, most us standard computers**

Continue

- **Parallel processors can improve the training and running of neural networks**

- **Acceleration boards expedite the computational work of neural computers**

- **Neural computing excels in pattern recognition, learning, classification, generalization and abstraction, and interpretation of incomplete input data**

- **ANN do well at tasks done well by people and *not* so well at tasks done well by traditional computer systems**

Continue

- **Machine learning describes techniques that enable computers to learn from experience**
- **Machine learning is used in knowledge acquisition and in inferencing and problem solving**
- **Machines learn differently from people; usually not as well**
- **Neural networks are useful in data mining**
- **Neural networks can enhance DSS**

Continue

Questions for the Opening Vignette

1. **Describe why neural networks were used.**
2. **Why is historical data used?**
3. **Is the neural network really learning? Why or why not?**
4. **How could the neural network possibly outperform a human?**
5. **What related applications can you think of for using neural computing?**

Group Exercise

Self-organizing neural network class exercise. **Everyone in the class should stand in the front of the room, away from the desks and chairs (a room free of furniture is even better). Without making a sound everyone should line up in order by height, tallest to shortest. Then, at a signal line up alphabetically. Notice that only minimal information is really needed by each person to identify their position. Discuss how this relates to how self-organizing neural networks learn.**