Project Objectives :-

1. **Revolutionary Real Time Automated Trading Infrastructure Solution for Financial Institutions (traditional & online) [Focused on Futures and Options Markets]**

2. **Automation of an accounting\Sales\Production system in a company\industry.**

3. **Designing MIS for Decision Support System (DSS)**

4. **Website offering information about computerized accounts.**

System Analysis :-

The System:

- is fully automated
- contains real time execution module, risk control and full trading simulation
- defines and formulizes a Virtual Trader
- utilizes artificial intelligence principles to adapt strategies to market changes
- is flexible and comprehensive

System Advantages :-

The System:

- opens up a window of opportunity for financial institutions to create a new online market
- is non intrusive
- is scalable
- is easily adaptable to local markets
- is fully automated, immediate (real time) and accurate
- handles massive numbers of executions

Artificial Intelligence in Finance

Introduction

Business sectors ranging from banking and insurance to retail are benefiting from a whole new generation of "intelligent" computing techniques. Successful applications include asset forecasting, credit evaluation, fraud detection, portfolio optimization, customer profiling, risk assessment, economic modeling, sales forecasting and retail outlet location.

The techniques include expert systems, rule induction, fuzzy logic, and neural networks and genetic algorithms, which in many cases are outperforming traditional statistical approaches. Their essential features include the ability to recognize and classify patterns, learning from examples, generalization; logical reasoning from premises, adaptability, and the ability to handle data, which is incomplete, imprecise, and noisy.

Background

There are two general classes of artificial intelligence techniques that have found an application in finance:

- ***Expert_Systems***
  These systems use knowledge-based or rule-based reasoning to make decisions about a market. Typically, a sufficiently large knowledge base of financial expertise is developed and is used to assist a real human trader in making decisions.
- ***Neural_Networks***
  These systems are comprised of a connectionist network and a surrounding body of non-linear techniques, which are used to predict new market data from historical databases. Usually, these systems start with no inherent knowledge of their own, they instead build behaviors that predict the market, by a system of trial and error.

Expert Systems

The purpose of this class of system is, usually, to render intelligent advice to an investor, by consulting an internal knowledge base.

A good example of this is a system called **TARA (Technical Analysis and Reasoning Assistant)**.

Specification :-

TARA was developed at Manufacturers Hanover Trust, and its basic goal is to give real-time advisement to foreign currency traders.

TARA would know to bring up charts of that countries stock and currency index. TARA also could apply simple rules to a chart, and make some form of recommendation, usually a simple buy/hold/sell advisement based on the movement of certain figures in accordance with pre-defined rules.

TARA was generally characterized as a success by its reviewers and, after the initial deployment in 1987, several of the systems were ordered. However, despite its success, some difficulties did arise, many of which are indicative of larger problems with AI systems in the investment business.

TARA's development team had some initial hurdles developing a user interface that was strong enough to hold the interest of traders with differing styles. In addition, when it was first introduced, traders were very skeptical of the system although it had worked well for months on a "dry run." This reluctance illustrates a peculiar characteristic of AI in the investment business that seems ubiquitous: despite a system's success using previous or realistic data, traders will almost always refuse to accept it as a tool until it proves itself on-line with real market situations.

Limitation :-

In general, in finance, expert systems are limited to producing data that must later be interpreted by humans.

Neural Network Modeling

The most promising application of artificial intelligence in finance is the use of neural networks. In this class of utilization, a neural network attempts to predict stock values and make portfolio decisions. This type of modeling differs from the other applications that have been discussed in several ways:

- The various models are strictly quantitative – the work is based on a hope that there is some underlying pattern or demonstrable stochastic relationship between various market indicators and stock returns that, when discovered, can be exploited.

- The models are usually generalized to the point where human qualitative judgments are completely removed from the predictions, thus eliminating any biases that might be incorrect.

This is a stark contrast to the agent or expert system models, which attempt to quantify what is already known about the market – neural networks try to rediscover these relationships.

The thrust of economic AI research largely ignores the classical efficient market theory, in favor of the idea that extrapolation of past data can be an effective way to trace future value.

Although questionable in a highly reversible market, the central belief is that stock returns follow different patterns in different circumstances. The hope of this tact is that if nothing has undermined the past’s ability to predict the future, it should be possible to construct a statistical model <u>to produce accurate predictions of future stock performance from existing historical data</u>. The general methodology is to compile sufficiently large library of past patterns into a database, and apply a non-linear pattern recognition system to identify previously seen fluctuations as they newly arise in the market. Typically, a neural network is used as the <u>pattern recognizer</u>.

Neural networks are well suited to the application of time-series prediction because they can improve their performance on a particular task by trial and error. In addition, while they are mathematically complex, their implementation is simple and somewhat abstract. Unfortunately, neural networks have a tendency to over-fit the data, and, if not properly tuned, can spuriously react to coincidental patterns. With the application of a fuzzy logic decision-maker that rates the decisiveness of a given prediction, this brittleness can be suppressed. In addition, a common way to "tune" a neural network is to evolve it through a genetic algorithm whose selection criteria is the effectiveness of the network as a predictor.

Due to industry skepticism, the majority of AI systems that have been developed to take a very conservative angle in their actual implementations. AI techniques as a special option to existing systems. The research has also concluded that most financial data needs to be pre-processed in some way before being used in neural networks, and they typically use regressions or differences to filter their data so the network more effectively uses it. To deal with the aforementioned difficulties inherent in neural networks themselves, the actual forecasting systems breed the nets with genetic algorithms that start with a random population and select parents with the greatest percentage of correct decisions.

The final judgement on neural networks is difficult to develop. As mentioned before, no one is talking about how successful such systems really are. But neural systems are certainly out there, and there are a number of firms, like NTS and PanAgora that specialise in bringing AI systems into existing quantitative asset models. There may be something to be said for computer trading in general - because the biases a computer develops are entirely empirical to the data, they are in some ways immune to falsely acquired human judgements. Then again, since it is hard to tell exactly why a neural net makes the decisions it does, it is possible that over time the network will simply stop working as new situations arise. Clearly the next several years will show the ultimate value of these systems as they become more popular in the market and new techniques are developed for using them.

Conclusion

The discussion of AI in finance has generally indicated that artificial intelligence techniques are only partially useful to most applications in finance. Most systems still require a great deal of heuristic knowledge, which must be provided by experts who already understand many of the market's nuances. For the expert systems, this knowledge comes in the form of rules and knowledge bases. In neural network applications, it must be provided in the pre-processing and interpretation phases. It seems that such systems are best used as supplements to an existing team of experts instead of on their own, and hence, they are more in the realm of statistical tools than "Artificially Intelligent" agents. Nevertheless, they are powerful techniques and as our development of them progresses, it is likely that they will find greater and greater utilization on Wall Street.

AI Systems and Definitions

A good general definition of AI could be:

AI is the part of computer science concerned with designing intelligent computer systems, that is, computer systems that exhibit the characteristics we associate with intelligence in human behaviour - understanding language, learning, reasoning and solving problems.

There are **many other** suitable definitions -- See A theme we will develop in this course is that most AI systems can broken into:

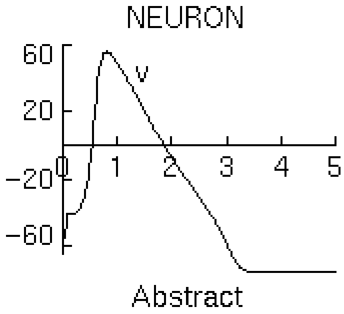
- Search,
- Knowledge Representation,
- applications of the above.

Search has been studied in great depth as follows :-

- Knowledge representation deals with finding a means of encoding knowledge so that a machine can use it.
- *e.g* Expert systems have to work with a knowledge base as do many other reasoning tasks.
- We will look at knowledge representation early on in this course.

- Tasks such a planning, reasoning, learning, understanding basically involve some searching and perhaps updating of a knowledge base.
- Tasks such as vision, natural language understanding, speech recognition and robot planning involve searching knowledge also. We will look at some of these tasks in the remainder of the course.

A 1 Neural networks, basic concepts



- **Description** of neural network and basic concepts.
- Fields of **application** of neural networks.

A 1 1 Description of neural network and basic concepts

Artificial neural networks (ANN) are parallel systems for information processing, inspired in the way in which biological neuron networks in the brain process information.

Everybody knows that the human brain is superior to a digital computer in many tasks. For example, in the processing of visual information: a one-year old child recognized objects, faces, ... better and faster than the best Artificial Intelligence system designed for such a task, even if it run on a supercomputer. Computers only exceed the human brain in tasks based mainly on simple arithmetic.

Due to the inspiration of ANN in the brain already mentioned, their main applications are focused on fields in which human intelligence cannot be emulated satisfactorily by arithmetical algorithms that can be implemented in computers. Furthermore, one expects that such ANN have characteristics similar to those of the brain:

- They will be robust and tolerate failure. Every day, a large number of neurons die in the brain without affecting appreciably its functioning.
- They will be flexible The brain adapts to new circumstances through learning.
- They will be able to deal with blurry, incomplete, probabilistic, noisy or inconsistent information.
- They will be highly parallel. The brain is formed by a lot of neurons interconnected with one another and it is precisely their collective behavior what characterizes their way of processing information.

The key point of ANN is the new structure of these systems for the processing of information. They are composed, as in the brain, of a very large number of basic elements (neurons) highly interconnected among one another with an answer model for each element depending on the environment and are very similar to the behavior of biological neurons. These models are simulated in conventional computers and this collective behavior of all the elements grants them these peculiar characteristics for the resolution of complex problems. ANNs, as people, learn from examples. Learning in biological systems entails the modification of the interconnectivity among neurons and this is also true for ANNs.

ANNs have been applied to an increasing number of considerably complex real problems, as for example, the recognition of models, data classification, predictions, etc. Their most important advantage lies in solving problems which are too complex for conventional techniques: problems which do not have a specific algorithm for their solution, or whose algorithm is too complex to be found.

In general, Artificial Neural Networks have been clearly accepted as very efficient systems for the treatment of information in many fields. This has resulted in a variety of commercial applications (in products as well as in services) of this technology of neural networks.

A 1 2 Fields of application of neural networks

The fields of application of neural networks are normally those in which statistical and/or linear models can be used or are used. In general, the use of neural networks provides much better results. Some fields in which neural networks are applied are:

- **Finance.**
  - Index prediction
  - Fraud detection
  - Credit risk, classification
  - Prediction of share profitability
- **Business**
  - Marketing
  - Cross sales
  - Sale campaigns
- **Treatment of text and form processing.**
  - Recognition of characters printed mechanically
  - Graphic recognition.
  - Recognition of hand-written characters.
  - Recognition of manual italic writing
- **Food**
  - Odor and aroma analysis.
  - Customer profiling depending on purchase.
  - Product development.
  - Quality control.
- **Energy.**
  - Electrical consumption prediction
  - Distribution of water resources for electrical production
  - Prediction of gas consumption
- **Manufacturing industry.**
  - Process control.
  - Quality control.
  - Control of robots.

- **Medicine and health**

    - Help to diagnosis.
    - Image analysis.
    - Medicine production.
    - Distribution of resources.

- **Science and Engineering.**

    - Data analysis and classification
    - Chemical engineering.
    - Electrical engineering.
    - Climatologic.

- **Transports and Communications.**

    - Route optimization.
    - Optimization of the distribution of resources

Application Areas

Application areas may include but are not limited to:

- Artificial Stock Markets
- Simulation of Social Processes
- Evolutionary Game and Industrial Organization
- Financial Engineering
- Financial Data Mining
- Trading Strategies
- Portfolio Management
- Derivative Pricing
- Term Structure Models
- Financial Time Series Forecasting and Analysis

Techniques:

Techniques may include, but are not limited to:

- Artificial Neural Networks
- Fuzzy Logic
- Evolutionary Strategies
- Evolutionary Programming
- Genetic Algorithms
- Genetic Programming
- Statistical Classifiers
- Decision Trees
- Inductive Logic Programming
- Self-organized Mapping
- Reinforced Learning
- Wavelet Analysis
- Rough Sets
- Support Vector Machines
- Hybrid Systems

B System requirements

Minimum requirements for software

- Any operating system based on 32 bit Windows from Windows 95 onwards.

Minimum requirements for hardware

- A compatible PC, from Pentium 133 MHz onwards with 32 MB of RAM.
- An installed operating system of 32 bits: Windows9x, Windows NT. Windows ME, Windows 2000, Windows XP
- A graphic card with at least 4 MB of RAM and a compatible monitor.
- Serial mouse, bus or PS2.
- Hard disk with at least 20 MB free space.
- A printer if you want to print out the results and diagrams.

Miscellaneous Financial AI sites (name + some info)

AI Finance and Investment Gemes :-:

is intended to provide short cuts for visiting places of interest to persons involved in applying AI to finance and investment problems.

AI On Wall Street :-

ABSTRACT: In the past five years, complex artificial intelligence systems have found new applications in the finance industry, an area dominated by traditional analytic techniques. This paper presents the current state of the art of artificial intelligence disciplines in finance. Three basic areas of application will be discussed. The first is the use of simulated interactive agents to explain the behavior of the market from a theoretical perspective. Secondly, knowledge-based expert systems are utilized to give traders real-time, market-level advice. The third topic, which is the most discussed and will receive special treatment here, is the implementation of neural networks and other advanced analytical techniques in time-series analysis and portfolio generation.

Applying Neural Networks to Market Timing :-

This course is designed to develop skills in building neural-network-based market timing systems. Theoretical insights into nonlinear dynamic systems, trading behavior, and the statistics of markets are used as background for the practice of building tradable systems. You will develop skills in using NeuralWorks Predict as well as NeuralWorks Professional II/PLUS. Previous experience with NeuralWare's products is helpful, but not necessary.

BrainMaker Neural Networks :-

BrainMaker lets you use your computer for business and financial forecasting, pattern recognition, medical diagnosis, sports handicapping...almost any activity where you need special insight. With more than 20,000 systems sold, BrainMaker is the world's best-selling software for developing neural networks.

Braincel :-

Promised Land Technologies has been at the leading edge of developing the tools with which anyone can take advantage of powerful Neural Net technology to develop forecasting applications and the ability to "mine" data for hidden links and relationships. Our flagship product is Braincel, an add-in to Excel for Windows 3.1 and up. A demo version of Braincel can be downloaded to try if you already have a version of Excel for Windows

Financial Business Application for Neural Networks :-

It takes you to a postscript document describing some of the common pitfalls and errors made when carrying out financial prediction with a neural network.

Forecast

Neural Networks for Financial & Economic Predictions :-

A number of economic and financial problems are investigated, from predicting the U.S. Index of Industrial Production to building a working portfolio management system.

The page for Artificial Intelligence in Finance :-

This page contains references and information regarding the use of artificial intelligence techniques in finance. In particular, this page presents information regarding the use Natural Language Processing technologies in finance as general support, explanatory and forecasting tool and the Durham financial information extraction system. The NLP technology is rather new to Finance and its importance widely unknown by the financial operators.

Neural Networks, Genetic Algorithms and Case-Based Reasoning for Finance :-

This tutorial shows how these quantitative techniques are used in practice, with topics covered including:

1) Comparing linear, nonlinear, and time-varying regression to neural networks,
2) Using neural network techniques to adaptively learn how to price options,
3) Using genetic algorithms in portfolio optimization problems,
4) Designing derivative securities with genetic algorithms,
5) The news on news:
how case-based expert systems integrate subjective knowledge into fundamental models.

BUSINESS APPLICATIONS OF NEURAL NETWORKS

The State-of-the-Art of Real-World Applications

Neural networks are increasingly being used in real-world business applications and, in some cases, such as fraud detection, they have already become the method of choice. Their use for risk assessment is also growing and they have been employed to visualise complex databases for marketing segmentation. This boom in applications covers a wide range of business interests — from finance management, through forecasting, to production. The combination of statistical, neural and fuzzy methods now enables direct quantitative studies to be carried out without the need for rocket-science expertise.

This reviews the state-of-the-art in current applications of neural-network methods in three important areas of business analysis. It introduces new users to the potential and pitfalls of this new technology.

Financial Forecasting

Financial Forecasting describes the process by which firms think about and prepare for the future. The forecasting process provides the means for a firm to express its goals and priorities and to ensure that they are internally consistent. It also assists the firm in identifying the asset requirements and needs for external financing.

For example, the principal driver of the forecasting process is generally the sales forecast. Since most Balance Sheet and Income Statement accounts are related to sales, the forecasting process can help the firm assess the increase in Current and Fixed Assets which will be needed to support the forecasted sales level. Similarly, the external financing which will be needed to pay for the forecasted increase in assets can be determined.

Firms also have goals related to Capital Structure (the mix of debt and equity used to finance the firms assets), Dividend Policy, and Working Capital Management. Therefore, the forecasting process allows the firm to determine if its forecasted sales growth rate is consistent with its desired Capital Structure and Dividend Policy.

The forecasting approach presented in this section is the Percentage of Sales method. It forecasts the Balance Sheet and Income Statement by assuming that most accounts maintain a fixed proportion of Sales. This approach, although fairly simple, illustrates many of the issues related to forecasting and can readily be extended to allow for a more flexible technique, such as forecasting items on an individual basis.

A Personal Finance "Spreadsheet"

1. Introduction

One of the critical requirements for the adoption of AI technology as a standardized human-computer interface is the creation of a core set of application programming interfaces. For AI to become the primary mode of user interaction, you must have interfaces to a collection of common applications such as email, database, address book, calculator, and spreadsheet programs.

This report describes a first pass at the "spreadsheet" application for managing personal finances.

2. Sample Dialogue

The goal of this experiment is to produce a believable natural language interface to a trivial personal finance system. The following sample dialogue illustrates the flavor of conversations the bot should be able to handle:

```
Client: My checking account balance is 2300.
Robot: I got it. Do you have any more transactions?
Client: My savings account balance is 1000.
Robot: I got it. Anything else I can do for you?
Client: My visa card balance is 2100.
Robot: I got it. Thanks for the information.
Client: My citibank balance is 9000.
Robot: I got it. Is that it?
Client: I have 200 dollars in my wallet.
Robot: Alright. Thanks for the information.
Client: I spent 10 on a taxi.
Robot: I got it. Anything else I can do for you?
Client: I charged 23 for lunch on my visa card.
Robot: I got it. Thanks for the information.
Client: My landlord is mr bill.
Robot: Rent is theft!
Client: No kidding my rent is 1410 a month.
Robot: Would I kid you? 1410 is the market rate.
Client: I paid the rent with check number 101.
Robot: I got it. Thanks for the information.
Client: what is my checking account balance.
Robot: Your checking balance is 890.
Client: I charged 120 on my Visa card for the hotel.
Robot: Ok. Anything else I can do for you?
Client: What is my Visa card balance.
Robot: Your visa balance is -2243.
Client: How much money do I have in my wallet?
Robot: Your wallet balance is 190.
Client: I bought a book for 10.
Robot: Yes. I have recorded your transaction.
Client: now how much is in my wallet?
Robot: Why now? Your wallet balance is 180.
```

3. The transactions

The preceding dialogue reduces to the transactions listed in this section.

The **DEBIT** and **CREDIT** transactions have exactly the same format, except that the **DEBIT** transaction has a payee where the **CREDIT** has a payer. In either case the program stores the transactions in a comma- delimited format, making for easy import into other financial applications.

Specifically, the transaction record format is a comma delimited string of the form:

```
DEBIT account, amount, date, number, payee, category, note, id.
```

or

```
CREDIT account, amount, date, number, payer, category, note, id.
```

The account field identifies the account as checking, savings, wallet or the name of another account. The amount is indicated in dollars, as a nonnegative integer. The date includes a time stamp to the resolution of seconds. The number field usually indicates a check number or transaction number, if any. Payer/payee indicates the source/destination accounts for the transaction. The category field helps create a budget report later, showing how our income and expenses break down by categories such as rent, medicine, food, utility bills and so on.

The "security" of the system is limited to recording the client id, in this case local host, local domain. If the boot is online, care should be taken so that unauthorized clients cannot initiate personal financial transactions.

The records do not include a unique identifier or row number. Uniqueness of the records is only guaranteed by the timestamp, using the <date/> function (*ref. AIML*). This is fine if the transaction rate is always less than one per second. But a more realistic application may also have to include record identifiers.

The most sophisticated language processing demonstrated in this experiment was the use of AI predicates for "landlord" and "rent". This makes possible the transformation of sentences such as "I paid the rent with check number 101" into a transaction.

```
INITIALIZE BANK checking, 2300, Tue Dec 25 08:32:16 PST 2001,
localhost.localdomain.
CREDIT checking, 2300, tue dec 25 08 32 16 pst 2001, none, unknown,
unknown, none, localhost.localdomain.
INITIALIZE BANK savings, 1000, Tue Dec 25 08:32:24 PST 2001,
localhost.localdomain.
CREDIT savings, 1000, tue dec 25 08 32 24 pst 2001, none, unknown,
unknown, none, localhost.localdomain.
INITIALIZE CREDT visa card, 2100, Tue Dec 25 08:32:32 PST 2001,
localhost.localdomain.
DEBIT visa card, 2100, tue dec 25 08 32 32 pst 2001, none, unknown,
unknown, none, localhost.localdomain.
INITIALIZE CREDT citibank, 9000, Tue Dec 25 08:32:51 PST 2001,
localhost.localdomain.
DEBIT citibank, 9000, tue dec 25 08 32 51 pst 2001, none, unknown,
unknown, none, localhost.localdomain.
INITIALIZE BANK wallet, 200, Tue Dec 25 08:33:01 PST 2001,
localhost.localdomain.
CREDIT wallet, 200, tue dec 25 08 33 01 pst 2001, none, unknown, unknown,
none, localhost.localdomain.
DEBIT wallet, 10, tue dec 25 08 33 23 pst 2001, none, a taxi, unknown,
none, localhost.localdomain.
DEBIT visa, 23 for lunch, tue dec 25 08 33 34 pst 2001, none, unknown,
unknown, none, localhost.localdomain.
DEBIT checking, 1410, tue dec 25 08 34 07 pst 2001, number 101, mr bill,
rent, none, localhost.localdomain.
DEBIT visa, 120, tue dec 25 08 35 00 pst 2001, none, unknown, the hotel,
none, localhost.localdomain.
DEBIT wallet, 10, tue dec 25 08 35 28 pst 2001, none, unknown, a book,
none, localhost.localdomain.
```

4. The AIML Categories

The dialogue in section 2 resulted from the contents of the ALICE brain, plus the following set of "spreadsheet" categories. The basic idea is to reduce every financial statement into a transaction of the form **CREDIT**, **DEBIT** or **INITIALIZE**. There are two types of accounts, **CREDIT** accounts, mainly for **CREDIT** cards, and **BANK** accounts, which can include wallets, piggybanks, safes and other money containers, as well as regular bank **DEBIT** accounts like checking and savings.

The purpose of the **INITIALIZE** statement is to set an initial balance, open a new account, or override the spreadsheet's calculated balance for an account. **CREDIT** accounts are opened with a negative balance, initiated with a **DEBIT** transaction. **BANK** accounts are opened with a positive balance, initiated using a **CREDIT** transaction.

All of the **DEBIT** and **CREDIT** transactions are recorded in the default gossip file. This may not be the best place, but it made for a simpler pedagogical example. The program balance.sh calculates the account balances as need (see next section).

The following AIML categories represent a first-pass attempt to create the simple "spreadsheet". Not a true double-entry bookkeeping system, the categories nonetheless demonstrate the feasibility of building a personal finance application in AIML.

```
<aiml>
    <category>
        <pattern>ACKNOWLEDGE TRANSACTION</pattern>
        <template>
            <random>
                <li>Ok.</li>
                <li>Yes.</li>
                <li>Alright.</li>
                <li>I got it.</li>
            </random>
            <random>
                <li>Is there anything else I can help you with?</li>
                <li>Do you have any more transactions?</li>
                <li>I have recorded your transaction.</li>
                <li>Anything else I can do for you?</li>
                <li>Thanks for the information.</li>
                <li>Is that it?</li>
            </random>
        </template>
    </category>
    <category>
        <pattern>CREDIT ACCT * AMT * DATE * NUMBER * PAYER * CATEGORY *
NOTE *</pattern>
        <template>
            <srai>ACKNOWLEDGE TRANSACTION</srai>
            <think>
                <gossip>CREDIT <star/>, <star index="2"/>, <star
index="3"/>, <star index="4"/>, <star index="5"/>, <star index="6"/>,
<star index="7"/>, <id/>.</gossip>
            </think>
        </template>
    </category>
    <category>
        <pattern>DEBIT ACCT * AMT * DATE * NUMBER * PAYEE * CATEGORY *
NOTE *</pattern>
        <template>
            <srai>ACKNOWLEDGE TRANSACTION</srai>
            <think>
                <gossip>DEBIT <star/>, <star index="2"/>, <star
index="3"/>, <star index="4"/>, <star index="5"/>, <star index="6"/>,
<star index="7"/>, <id/>.</gossip>
            </think>
        </template>
    </category>
    <category>
        <pattern>INITIALIZE BANK ACCT * AMT *</pattern>
        <template>
            <srai>ACKNOWLEDGE TRANSACTION</srai>
            <think>
                <gossip>INITIALIZE BANK <star/>, <star index="2"/>,
<date/>, <id/>.</gossip>
                <srai>
                    CREDIT ACCT <star/> AMT <star index="2"/> DATE
<date/>
                </srai>
            </think>
        </template>
    </category>
    <category>
        <pattern>INITIALIZE CREDT ACCT * AMT *</pattern>
        <template>
            <srai>ACKNOWLEDGE TRANSACTION</srai>
            <think>
                <gossip>INITIALIZE CREDT <star/>, <star index="2"/>,
<date/>, <id/>.</gossip>
                <srai>
                    DEBIT ACCT <star/> AMT <star index="2"/> DATE <date/>
                </srai>
            </think>
        </template>
    </category>
    <category>
        <pattern>BALANCE ACCT *</pattern>
        <template>
            <system>sh /home/alicebot/balance.sh <star/>
            </system>
        </template>
    </category>
    <category>
        <pattern>HOW MUCH IS IN MY WALLET</pattern>
        <template>
            <srai>BALANCE ACCT WALLET</srai>
        </template>
    </category>
    <category>
        <pattern>HOW MUCH MONEY _ MY WALLET</pattern>
        <template>
            <srai>BALANCE ACCT WALLET</srai>
        </template>
    </category>
    <category>
        <pattern>HOW MUCH CASH _ I HAVE</pattern>
        <template>
            <srai>BALANCE ACCT WALLET</srai>
        </template>
    </category>
    <category>
        <pattern>WHO IS MY LANDLORD</pattern>
        <template>
            <get name="landlord"/>
        </template>
    </category>
    <category>
        <pattern>WHAT IS MY RENT</pattern>
        <template>
            <get name="rent"/> per month.</template>
    </category>
    <category>
        <pattern>MY RENT IS *</pattern>
        <template>
            <set name="rent"><star/></set> is the market rate.
        </template>
    </category>
    <category>
        <pattern>MY RENT IS * DOLLARS PER MONTH</pattern>
        <template>
            <srai>MY RENT IS <star/></srai>
        </template>
    </category>
    <category>
        <pattern>MY RENT IS * PER MONTH</pattern>
        <template>
            <srai>MY RENT IS <star/></srai>
        </template>
    </category>
    <category>
        <pattern>MY LANDLORD IS *</pattern>
        <template>
            <think>
                <set name="landlord">
                    <person/>
                </set>
            </think>
            Rent is theft!
            </template>
    </category>
    <category>
        <pattern>MY CHECKING ACCOUNT BALANCE IS *</pattern>
        <template>
            <srai>INITIALIZE BANK ACCT checking AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>MY CITIBANK BALANCE IS *</pattern>
        <template>
            <srai>INITIALIZE CREDT ACCT citibank AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>MY SAVINGS ACCOUNT BALANCE IS *</pattern>
        <template>
            <srai>INITIALIZE BANK ACCT savings AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>MY VISA CARD BALANCE IS *</pattern>
        <template>
            <srai>INITIALIZE CREDT ACCT visa card AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>I HAVE * IN MY WALLET</pattern>
        <template>
            <srai>INITIALIZE BANK ACCT wallet AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>I HAVE * DOLLARS IN MY WALLET</pattern>
        <template>
            <srai>INITIALIZE BANK ACCT wallet AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>WHAT IS MY * ACCOUNT BALANCE</pattern>
        <template>
            <srai>BALANCE ACCT <star/></srai>
        </template>
    </category>
```

```
    <category>
        <pattern>I HAVE * DOLLARS TO MY NAME</pattern>
        <template>
            <srai>INITIALIZE BANK ACCT wallet AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>I HAVE * DOLLARS</pattern>
        <template>
            <srai>INITIALIZE BANK ACCT wallet AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>WHAT IS MY * CARD BALANCE</pattern>
        <template>
            <srai>BALANCE ACCT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>DEPOSIT * IN CHECKING ACCOUNT</pattern>
        <template>
            <srai>CREDIT ACCT checking AMT <star/></srai>
        </template>
    </category>
    <category>
        <pattern>I PAID MY RENT WITH CHECK *</pattern>
        <template>
            <srai>
                DEBIT ACCT checking AMT <get name="rent"/> DATE <date/>
NUMBER <star/> PAYEE <get name="landlord"/> CATEGORY RENT
            </srai>
        </template>
    </category>
    <category>
        <pattern>I PAID THE RENT WITH CHECK *</pattern>
        <template>
            <srai>
                DEBIT ACCT checking AMT <get name="rent"/> DATE <date/>
NUMBER <star/> PAYEE <get name="landlord"/> CATEGORY RENT
            </srai>
        </template>
    </category>
    <category>
        <pattern>I CHARGED * DOLLARS ON MY ATM AT * FOR *</pattern>
        <template>
            <srai>
                DEBIT ACCT checking AMT <star/> DATE <date/> NUMBER none
PAYEE <star index="2"/> CATEGORY <star index="3"/>
            </srai>
        </template>
    </category>
</aiml>
```

The remainder of the AIML categories have been placed due to considerations of length.

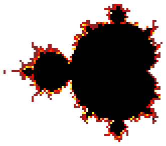
5. Conclusion

We have demonstrated the feasibility of a simple personal finance application with AIML. Despite its limitations, the program is adequate for a small set of personal finance needs.

To make this application more robust we should:

1. Create a realistic double-entry bookkeeping system.
2. Add a unique identifier to each transaction.
3. Integrate the program more closely with a real financial application.
4. Develop an expanded set of AIML patterns to capture a wider variety of transaction statements, including dollars and cents, international currencies, interest payments, and taxes.
5. Enhance the security feature to prohibit unauthorized clients from making personal financial transactions.
6. Substitute voice for text input. This is the kind of application you want to carry around with you, so you can utter "I spent 10 on a taxi", at the same time you pay the driver, and "I took out 100 from the ATM", while you are still at the cash machine.

More generally, we need to work on AIML APIs for a variety of everyday applications. These include, but are not limited to, address books, spreadsheets, databases, email, web browsers, word processors, and games. In the dream world of the future, the talking Star- Trek/HAL-style computer will handle all of the applications we use now on WIMP (Windows, Icons, Menus, Pointing Device) computers.



genetic algorithms

“**Natural selection is a mechanism for generating an exceedingly high degree of improbability** “ - Ronald Aylmer Fisher

Heuristics for cardinality constrained portfolio optimization ->

by Chang, Meade and Beasley

It presents three heuristic algorithms based upon genetic algorithms, tabu search and simulated annealing for finding the cardinality constrained efficient frontier.

It shows that constraints on downside risk lead to optimal asset allocations which differ from the mean-variance optimum. They also introduce constraints restricting the trading variables to be integers, on the holding size of assets and on the maximum number of different assets in the portfolio and use an optimization heuristic called threshold accepting.

It investigates the application of two heuristic methods, genetic algorithms and tabu/scatter search, to the optimisation of realistic portfolios. The model is based on the classical mean-variance approach, but enhanced with floor and ceiling constraints, cardinality constraints and nonlinear transaction costs which includes a substantial illiquidity premium, and is then applied to a large 100-stock portfolio.

Another application is index tracking. Portfolio selection using genetic algorithms and quadratic programming techniques to find both their performance and the proportion of the available capital that should be invested in each member company.

S/W :

Crystall Ball is an integrated software suite for risk analysis and what-if exercises. The Optquest module in the professional version is a tabu/scatter search procedure. Crystal Ball is a highly integrated and professional package.

How does Crystal Ball work?

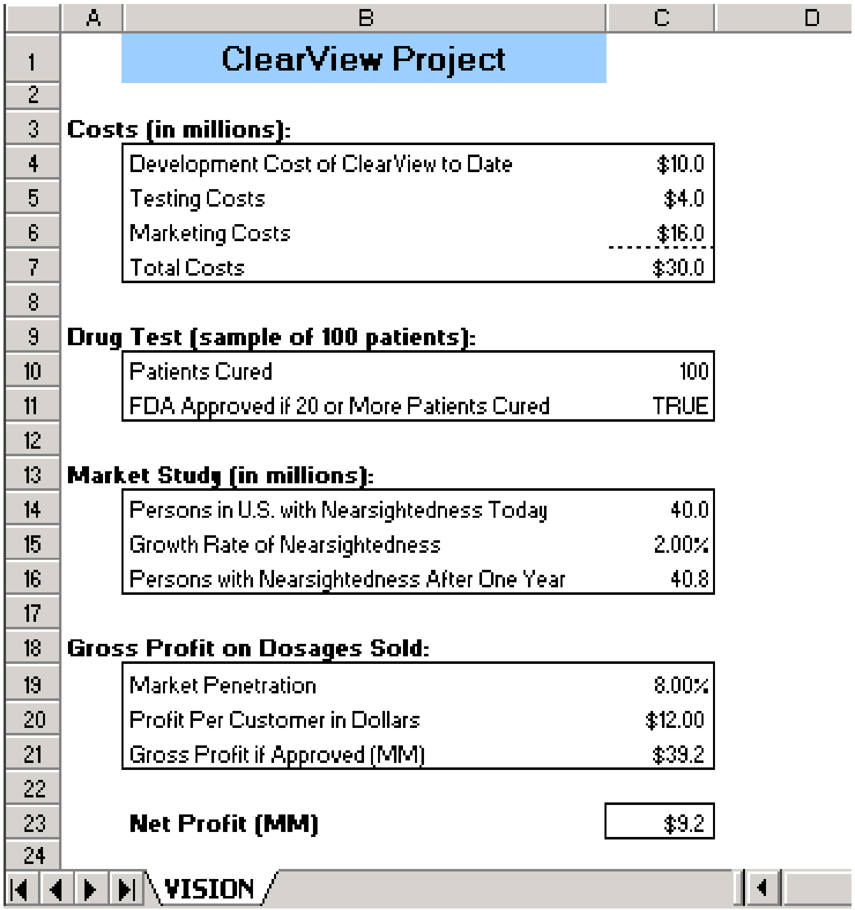
Crystal Ball 2000 Standard is an easy-to-use simulation program that helps you to analyze the risks and uncertainties associated with your Microsoft Excel spreadsheet models. This Spotlight is a quick overview of how Crystal Ball is used (it may take a little more than a minute!)

Excel models are **deterministic**, which means that the inputs are fixed (one value to one cell). You can only see one solution at a time. If you want to view alternative results, you need to manually change the inputs in the model.

Simulation is a way to quickly generate and analyze many possible results. Excel by itself cannot run simulations, so you need an add-in program like Crystal Ball to make Excel do this.

It Takes a Spreadsheet to Make a Model

Because Crystal Ball is an analysis tool, you can use it to simulate existing or new spreadsheets in any industry and for any application. For example, if you were a researcher in the pharmaceutical industry, you might develop the spreadsheet below to analyze the financial success of your project.



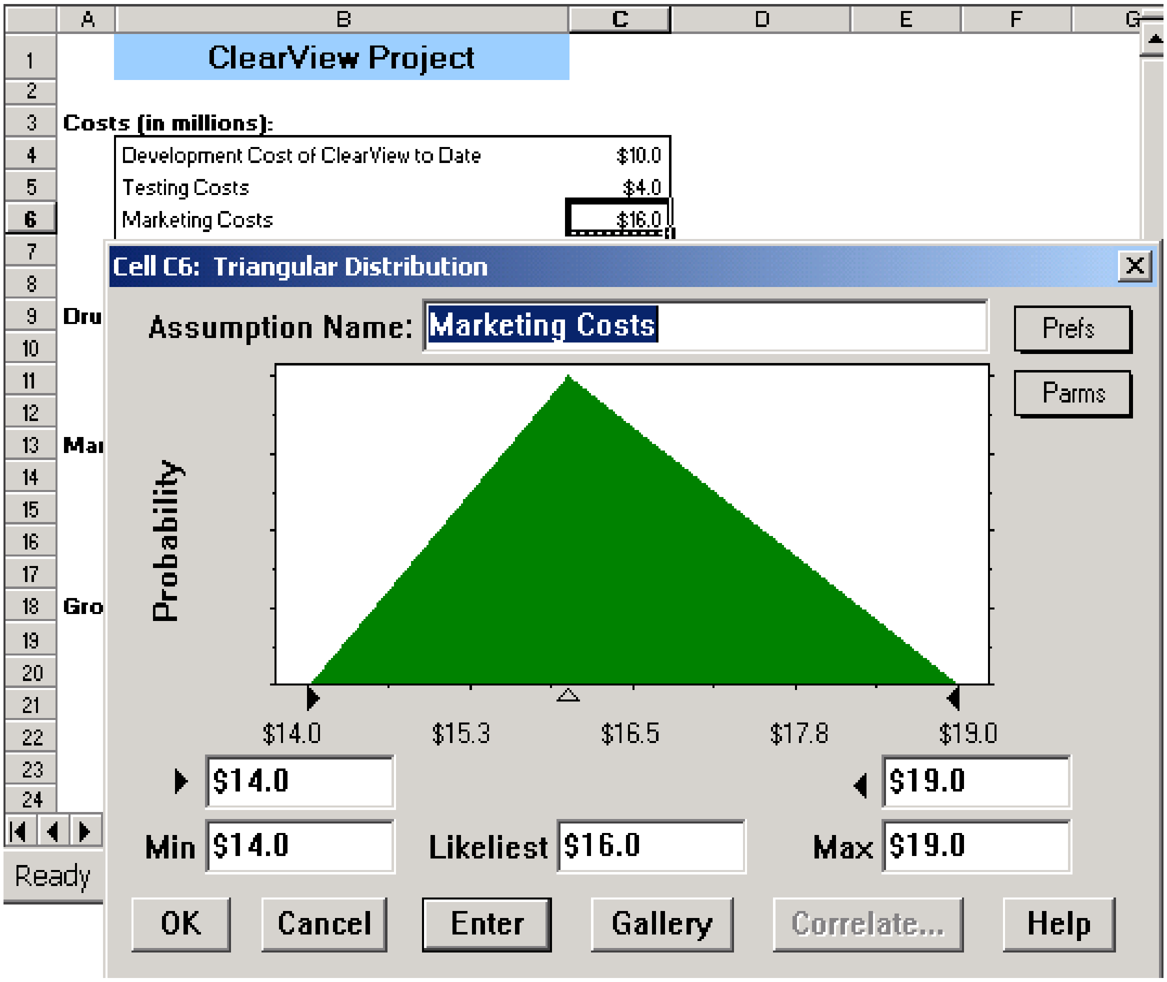
	A	B	C	D
1		ClearView Project		
2				
3	Costs (in millions):			
4		Development Cost of ClearView to Date	$10.0	
5		Testing Costs	$4.0	
6		Marketing Costs	$16.0	
7		Total Costs	$30.0	
8				
9	Drug Test (sample of 100 patients):			
10		Patients Cured	100	
11		FDA Approved if 20 or More Patients Cured	TRUE	
12				
13	Market Study (in millions):			
14		Persons in U.S. with Nearsightedness Today	40.0	
15		Growth Rate of Nearsightedness	2.00%	
16		Persons with Nearsightedness After One Year	40.8	
17				
18	Gross Profit on Dosages Sold:			
19		Market Penetration	8.00%	
20		Profit Per Customer in Dollars	$12.00	
21		Gross Profit if Approved (MM)	$39.2	
22				
23		Net Profit (MM)	$9.2	
24				

Without simulation, you calculate a Net Profit of $9,200,000. Not bad, but you have no sense of the likelihood of this result. Are you 75% sure this will happen? 100% sure? A simulation can show you the probability of a given outcome.

Identifying Uncertainty in Your Model and Defining Assumptions

The first step to using Crystal Ball is to determine which model inputs are uncertain. Which values are estimates? Which are averages? Once you have identified these, you use your knowledge of the uncertainty around the input to create a probability distribution for that cell (what we call an **assumption**). Crystal Ball lets you define these distributions.

In this case, you know Marketing Costs (cell C6) can vary between $14,000,000 and $19,000,000, but are most likely to be $16,000,000. You then use Crystal Ball to define a Triangular distribution with these parameters, as shown below.



ClearView Project

Costs (in millions):

Development Cost of ClearView to Date	$10.0
Testing Costs	$4.0
Marketing Costs	$16.0

Cell C6: Triangular Distribution

Assumption Name: Marketing Costs

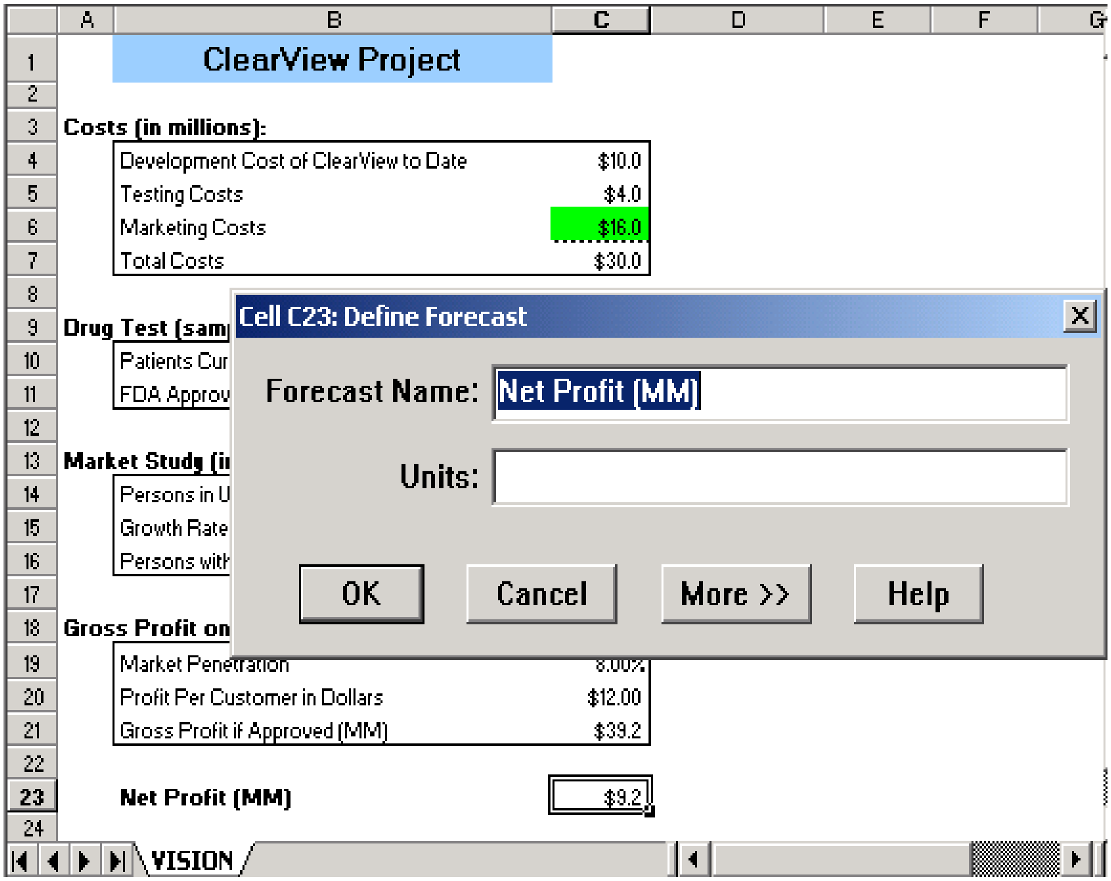
$14.0 | $19.0

Min $14.0 | Likeliest $16.0 | Max $19.0

The width of the triangle represents the range of possible costs, and the height of the triangle represents the likelihood of the value actually happening. The highest point of the triangle is $16,000,000, the most likely value.

Identify Which Forecasts You Want to Analyze

The next step is to identify a **forecast**. A forecast is a formula cell that you want to measure and analyze. In this model, you select the Net Profit (cell C23).



	A	B	C
1		ClearView Project	
2			
3	Costs (in millions):		
4		Development Cost of ClearView to Date	$10.0
5		Testing Costs	$4.0
6		Marketing Costs	$16.0
7		Total Costs	$30.0
8			
9	Drug Test (sam		
10		Patients Cur	
11		FDA Approv	
12			
13	Market Study (i		
14		Persons in U	
15		Growth Rate	
16		Persons with	
17			
18	Gross Profit on		
19		Market Penetration	8.00%
20		Profit Per Customer in Dollars	$12.00
21		Gross Profit if Approved (MM)	$39.2
22			
23		Net Profit (MM)	$9.2
24			

Cell C23: Define Forecast

Forecast Name: Net Profit (MM)

Units:

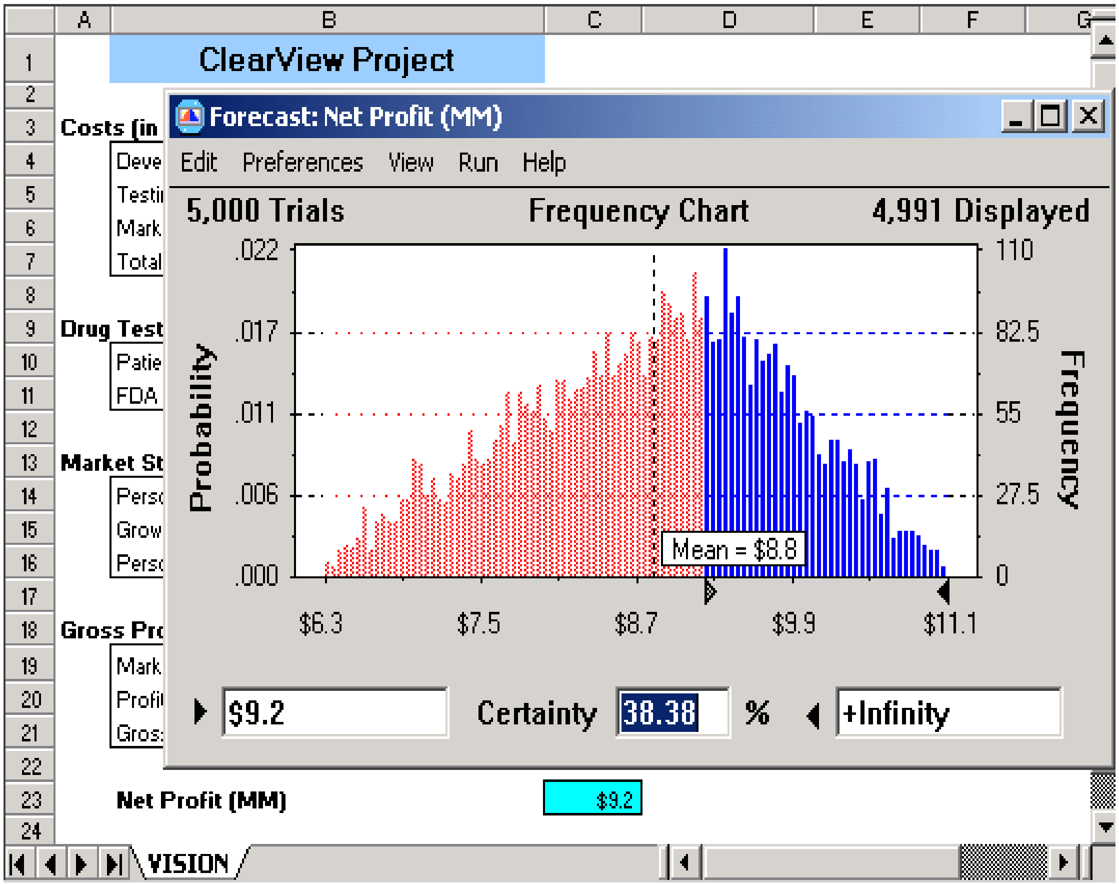
You can define multiple assumptions and forecasts. Once finished, you use the Crystal Ball command or toolbar icon to run a simulation. For each trial in this simulation, Crystal Ball enters a random value into the Marketing Cost cell based on the values you used to define the triangular distribution.

For trial #1, the random value might be $15,000,000, followed by $17,500,000 for trial #2, $16,875,000 for trial #3, and so on. Each time Crystal Ball enters a random value, it recalculates the spreadsheet and saves the forecast value in its memory for later analysis.

Analyzing Your Simulation Results

If you run a simulation for 5000 trials, then you have created 5000 forecasts (or possible outcomes), compared to the single outcome you started with in the deterministic spreadsheet. Simulation results are displayed in interactive histograms, or frequency charts. The chart below shows the results of 5000 trials of Net Profit.

(**NOTE:** the number of trials here was chosen arbitrarily. To run a more accurate number of trials, you would use the Precision Control Feature.)



5,000 Trials — Frequency Chart — 4,991 Displayed

$9.2 Certainty 38.38 % +Infinity

Net Profit (MM) $9.2

Note that the range of possible Net Profit values is $6,300,000 to $11,100,000, with a mean (average) value of $8,800,000. There is only a 38% certainty that you will receive a Net Profit of $9,200,000 as you originally predicted.

NeuroXL Predictor - Financial Forecasting Software Using Neural Networks

Neural Networks are ideal for financial prediction tasks such as currency price predictions. Currency prices are influenced by many different variables with are interrelated and behave in a non-linear fashion. **NeuroXL Predictor - Financial Forecasting Software** is a plug-in for Microsoft Excel that can be used to make accurate predictions of currency prices. The problem with traditional methods, such as regression, is that they make assumptions about the distribution of the underlying data, and often fail to recognize the interrelatedness of variables. NeuroXL Predictor, drawing on the latest in artificial intelligence technology, does not make false assumptions and recognizes even slight relationships between factors affecting prices.

About Neural Networks

Neural networks are a well-established technology for solving financial prediction problems. Loosely modeled after the human brain, neural networks are interconnected networks of independent processors, that by changing their connections (known as training), learn the solution to a problem. They are tolerant of imperfect data, and do not require formulas or rules.

NeuroXL Predictor as a tool for Financial Analysts

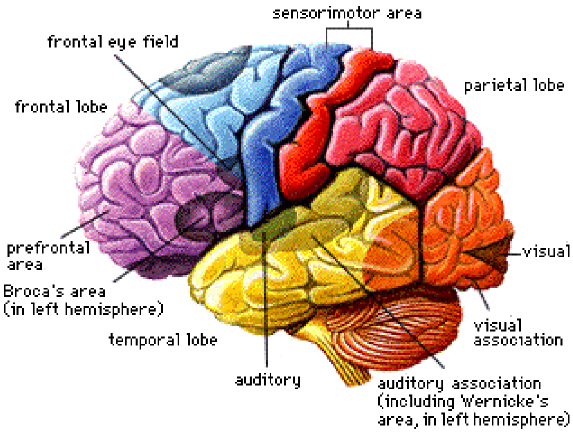
NeuroXL Predictor has become quite popular with financial analysts, as it has proven its predictive power through comparison with other applications and techniques. When the NeuroXL neural network approach is compared to other statistical techniques, such as regression and linear analysis, the predictive advantage of NeuroXL Predictor becomes clear. Using the application, all the user needs to do is to specify the inputs and outputs and set the required parameters. NeuroXL Predictor does the work of determining the relationship between variables and detecting relevant patterns in the data. Furthermore, since it is an add-in to Microsoft Excel, there is no need to import or export data. NeuroXL Predictor is a powerful and affordable solution for advanced estimation and forecasting. By harnessing the latest advances in artificial intelligence, it delivers accurate and fast predictions. Designed as an add-in to Microsoft Excel, it is easy to learn and use yet powerful enough for the most demanding professional.

APPLICATIONS IN BUSINESS:

- Business Forecasting & Classifying
- Demand Forecasting
- Marketing Forecasting & Segmenting
- Sales Forecasting

APPLICATIONS IN FINANCE:

- Credit & Loans
- Financial Forecasting
- Stock Forecasting
- Predict stock Prices

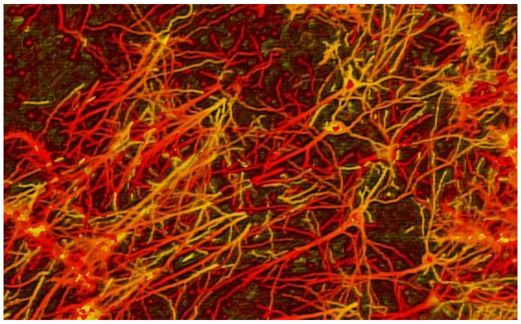


Neural Networks in the Brain

The brain is not homogeneous. At the largest anatomical scale, we distinguish **cortex**, **midbrain**, **brainstem**, and **cerebellum**. Each of these can be hierarchically subdivided into many **regions**, and **areas** within each region, either according to the anatomical structure of the neural networks within it, or according to the function performed by them.

The overall pattern of **projections** (bundles of neural connections) between areas is extremely complex, and only partially known. The best mapped (and largest) system in the human brain is the visual system, where the first 10 or 11 processing stages have been identified. We distinguish **feedforward** projections that go from earlier processing stages (near the sensory input) to later ones (near the motor output), from **feedback** connections that go in the opposite direction.

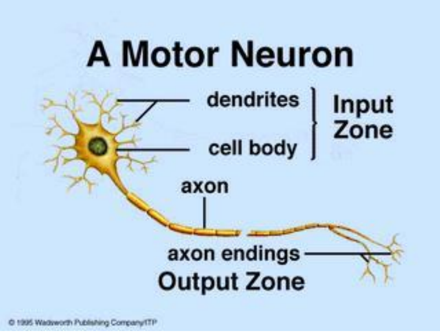
In addition to these long-range connections, neurons also link up with many thousands of their neighbours. In this way they form very dense, complex local networks:



Neurons and Synapses

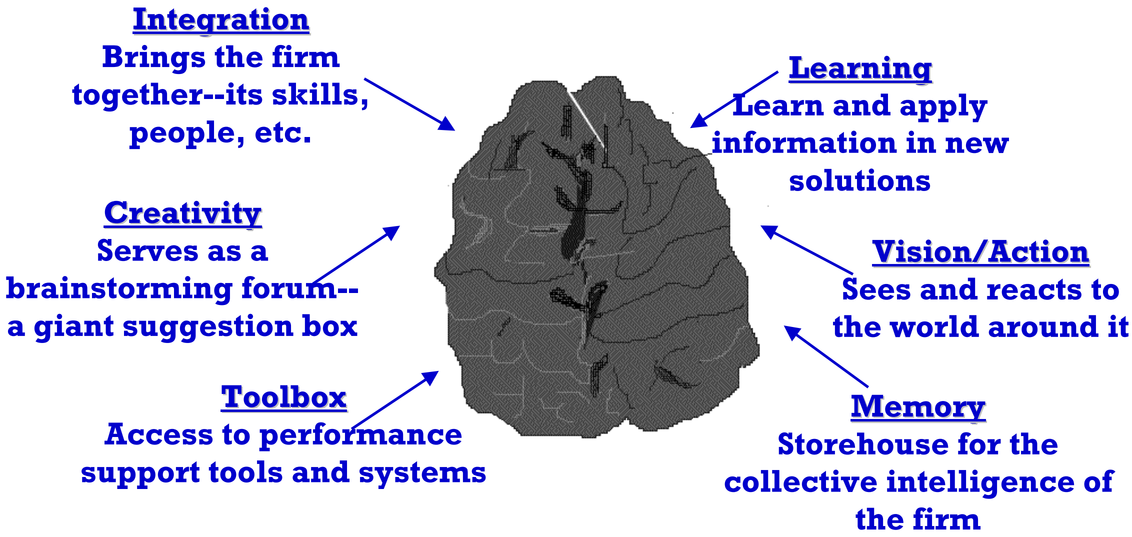
The basic computational unit in the nervous system is the nerve cell, or **neuron**. A neuron has:

- Dendrites (inputs)
- Cell body
- Axon (output)



A neuron receives input from other neurons (typically many thousands). Inputs sum (approximately). Once input exceeds a critical level, the neuron discharges a **spike** - an electrical pulse that travels from the body, down the axon, to the next neuron(s) (or other receptors). This spiking event is also called **depolarization**, and is followed by a **refractory period**, during which the neuron is unable to fire.

The axon endings (Output Zone) almost touch the dendrites or cell body of the next neuron. Transmission of an electrical signal from one neuron to the next is effected by **neurotransmittors**, chemicals which are released from the first neuron and which bind to receptors in the second. This link is called a **synapse**. The extent to which the signal from one neuron is passed on to the next depends on many factors, e.g. the amount of neurotransmittor available, the number and arrangement of receptors, amount of neurotransmittor reabsorbed, etc.



Integration
Brings the firm together--its skills, people, etc.

Learning
Learn and apply information in new solutions

Creativity
Serves as a brainstorming forum--a giant suggestion box

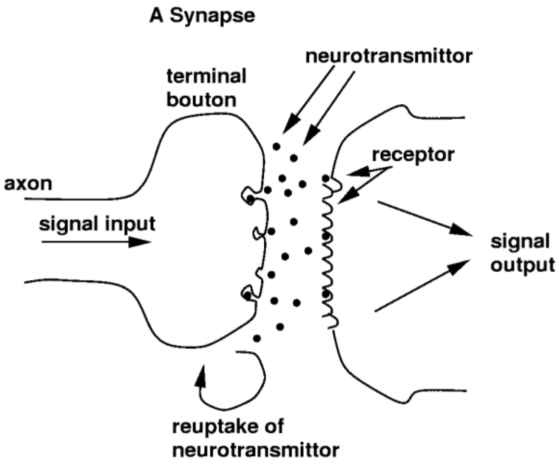
Vision/Action
Sees and reacts to the world around it

Toolbox
Access to performance support tools and systems

Memory
Storehouse for the collective intelligence of the firm

Ref: Knowledge management can be thought of as a virtual corporate brain — E-Learning

Synaptic Learning



Brains learn. Of course. From what we know of neuronal structures, one way brains learn is by altering the strengths of connections between neurons, and by adding or deleting connections between neurons. Furthermore, they learn "on-line", based on experience, and typically without the benefit of a benevolent teacher.

The efficacy of a synapse can change as a result of experience, providing both memory and learning through **long-term potentiation**. One way this happens is through release of more neurotransmitter. Many other changes may also be involved.

Long-term Potentiation:

An enduring (>1 hour) increase in synaptic efficacy that results from high-frequency stimulation of an afferent (input) pathway

Hebbs Postulate:

"When an axon of cell A... excites[s] cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells so that A's efficiency as one of the cells firing B is increased." Bliss and Lomo discovered LTP in the hippocampus in 1973

Points to note about LTP:

- Synapses become more or less important over time (plasticity)
- LTP is based on experience
- LTP is based only on *local* information (Hebb's postulate)

Summary

The following properties of nervous systems will be of particular interest in our neurally-inspired models:

- parallel, distributed information processing
- high degree of connectivity among basic units
- connections are modifiable based on experience
- learning is a constant process, and usually unsupervised
- learning is based only on local information

Performance degrades gracefully if some units are removed.