

Chapter 1

Artificial Intelligence

1.1 Artificial Intelligence and Intelligence

The goal of artificial intelligence is to try to develop computer programs, algorithms, and computer architectures that will behave very much like people and will do those things that in people would require intelligence, understanding, thinking, or reasoning. There are two important aspects to this study. First, there is the very grand goal of finding out how intelligence and human thinking works so that the same or similar methods can be made to work on a computer. This makes the subject on a par with physics where the goal is to understand how the whole universe of matter, energy, space, and time works. A second goal of AI is more modest: it is to produce computer programs that function more like people so that computers can be made more useful and so they can be made to do many things that people do and perhaps even faster and better than people can do them. These will be the problems that this book deals with, the grand aspect and the modest one.

1.1.1 Intelligence

To begin the consideration of *artificial* intelligence, it would be appropriate to start with some definition of *intelligence*. Unfortunately, giving a definition of intelligence that will satisfy everyone is not possible and there are critics who claim that there has been no intelligence evident in artificial intelligence, only some modestly clever programming. Thus, to begin this book, we must briefly delay looking at the normal sort of material that would be found in the first chapter of a textbook and instead look first at the controversy that surrounds the definitions of intelligence and artificial intelligence. Looking at this debate will not settle the issues involved to everyone's satisfaction and readers will be left to form their own opinions about the nature of intelligence and artificial intelligence. To begin looking at the definition of intelligence, we will start with aspects of intelligence where there is no disagreement and then move on to the issues that are hotly debated.

Everyone agrees that one aspect of human intelligence is the ability to respond correctly to a novel situation. Furthermore, in giving intelligence tests where the goal is to solve problems, people who quickly give the correct answer will be judged as more intelligent than people who respond more slowly. Then on a long test, the "smarter" or more "intelligent" people will get more correct answers than less smart, less intelligent people. Within this process there is an important aspect to consider. In order to be able to respond correctly to a novel situation, the situation cannot be too novel. Thus, if the situation at

hand is to do some calculus problems, you cannot expect people who have never done any calculus to manage to respond at all. Familiarity with the subject area is necessary to be able to demonstrate intelligence. Knowledge gained by experience is essential. Then you can look (or be?) more intelligent in a certain area simply by having more experience with that area.

The matter of possessing a certain amount of knowledge about a subject area can be quite subtle. For instance, adults ordinarily assume that it is easy to tell one person from the next simply by looking at them. It is assumed that adults have some kind of universal pattern recognition ability. However, it is often the case according to many media reports that when Americans visit some foreign country, especially, say, China, Americans often report that all the Chinese look the same. Of course, Chinese do not think that all Chinese look the same because native Chinese have had an extensive amount of experience recognizing Chinese faces. And to turn the tables, when Chinese students come to America they often report that all Americans look the same.¹ Thus, even a “simple” task like recognizing faces is not some kind of universal ability that adults develop but it is an ability that is developed to work within their own specific environment and which will not work very well outside that environment.

In addition to knowledge, speed, and experience, another key element of intelligence is the ability to learn. Everyone agrees that an intelligent system must be able to learn since obviously any person or program that cannot learn or which “mindlessly” keeps repeating a mistake over and over again will seem stupid. In fact, since as people learn a new task they get faster and faster at it, some people might require programs to get faster and faster as well.

If intelligence consisted of only storing knowledge, doing pattern recognition, solving problems, and the ability to learn, then there would not be any problem in saying that programs can be intelligent. But there are other qualities that some critics believe are necessary for intelligence. Some of them are intuition, creativity, the ability to think, the ability to understand or to have consciousness, and feelings. Needless to say it is hard to pin down many of these vague quantities, but this has not stopped artificial intelligence researchers and critics of AI from debating the points ad infinitum. Now we will mention some of the more prominent arguments.

1.1.2 Thinking

The issue of whether or not a machine could think might be decided quite easily by determining *exactly* how people think and then showing that the machine operates internally the same way or so close to the same way that there is no real difference between a human thinker and a machine thinker. For instance, some AI researchers have proposed that thinking consists of manipulating large numbers of rules, so if that is all that a person does and the machine does the same thing, it too, should be regarded as thinking. Or, for another example, it has been suggested that thinking in people involves quantum mechanical processing. If this is the case, then an ordinary computer could not think but it is always possible that the right kind of quantum mechanical computer could think. Settling the issue this way may be simple, but it will be a long time before we know enough about human

¹This comes from an informal survey by the author of Chinese students.

thinking to settle it this way. In the meantime, some people have proposed a weaker test for thinking: the Turing test.

1.1.3 The Turing Test for Thinking

Turing [238] and his followers believe that if a machine behaves very much like a person who is thinking, then the term thinking should apply to what the machine is doing as well. People who argue the validity of this test believe it is the running of an algorithm on a computer that constitutes thinking and it should not matter whether the computer is biological or electronic. This viewpoint is called the *strong AI* viewpoint. On the other hand, people who believe that electronic computing can only simulate thinking are said to have the *weak AI* viewpoint.

The most common version of the Turing test is the following (for Turing's original version, see Exercise 1.1): Put a person or a sophisticated computer program designed to simulate a person in a closed room. Give another person a teletype connection to the room and let this person interrogate the occupant of the closed room. The interrogator may ask the occupant any sort of question, including such questions as, "Are you human?" "Tell me about your childhood." "Is it warm in the room?" "How much is 1087567898 times 176568321?" In this last question a digital computer has a decided advantage over a human being in terms of speed and accuracy so that the designers of the simulated human being must come up with a way to make it as slow and unreliable as people are at doing arithmetic. In the case of "Are you human?" the machine must be prepared to lie. It is given, of course, that if the occupant of the sealed room is a person, the person is thinking. If after a short period of time the questioner could be fooled into thinking that the occupant was a person when it actually was a machine, it should be fair to say that the machine must also be thinking.

With a sufficiently complex computer and computer program, it would be a virtual certainty that *many* naive questioners will be unable to determine after a *short* period of time whether or not the occupant of the sealed room is a human being or a machine simulating a human being. However, it also seems a virtual certainty that more determined and sophisticated questioners will find ways to tell the difference between a machine and a human being in the sealed room (for instance see Exercise 1.1).

Notice also that the Turing test is relatively weak in that to a large extent it is a test of knowledge: if a computer failed to pass the Turing test *because it did not know something that a human being should know* it is no reason to claim that it is not thinking! Thinking is something that is independent of knowledge.

1.1.4 The Chinese Room Argument

An important argument against the strong AI viewpoint is the Chinese room argument of Searle [196, 197]. In this thought experiment the occupant of the Turing test room has to communicate in Chinese with the interrogator and Searle modifies the Turing test in the following way. Searle goes into the closed room to answer questions given to him despite the fact that he does not know any Chinese. He takes with him into the room a book with a Chinese understanding algorithm in it plus some scratch paper on which to do calculations. Searle takes input on little sheets of paper, consults the book that contains the algorithm for

understanding Chinese, and by following its directions he produces some output on another sheet of paper. We assume that the output is good enough to fool almost anyone into thinking that the occupant of the Chinese room understands the input and therefore must be thinking. But Searle, who does not understand any Chinese does not understand the input and output at all, so he could not be thinking or understanding. Thus merely executing an algorithm, even if it gets the right answers, should not constitute understanding or thinking.

Believers in strong AI then reply that while Searle does not understand, it is the whole room, including Searle, the algorithm in the book, and the scratch paper that is understanding. Searle counters this by saying he could just as well memorize the rules, do away with the pencil and paper, and do all the calculations in his head, but he still would not be understanding Chinese. Searle takes the point even farther by noting that a room full of water pipes and valves operated by a human being could, in principle, appear to understand without actually understanding as a real Chinese person would understand if that person was in the Turing test room.

The point of the argument is that merely executing some algorithm should not constitute understanding or thinking, understanding and thinking require something more. Searle supposes that the something more in people comes from having the right kind of hardware, the right kind of biology and chemistry.

1.1.5 Consciousness and Quantum Mechanics

Another criticism of the strong AI viewpoint is that intelligence, thinking, and understanding require consciousness. Of course, no one can give a solid definition of consciousness or a foolproof test for it. To the critics of strong AI, consciousness seems to be something that is orthogonal to computing, orthogonal to ordinary matter, but something that people and perhaps higher animals have. The strong AI position on consciousness is that it is something that will emerge in a system when a sufficiently complex algorithm is run on a sufficiently complex computer.

Recently, Roger Penrose, a mathematical physicist, has written two popular books [148, 149] giving his criticism of the strong AI viewpoint. He argues that intelligence requires consciousness and consciousness involves a nonalgorithmic element, an element that no ordinary computer running an algorithm can duplicate. Furthermore, according to Penrose, the nonalgorithmic element involves quantum mechanical effects. Lockwood [101], Wolf [263], and Nanopoulos [133] also speculate on how the mind might operate quantum mechanically and how consciousness might arise from quantum mechanical effects.

1.1.6 Dualism

The 17th century philosopher and mathematician, Rene Descartes, was a proponent of the idea that there is more to a human being than just plain matter, there is an additional component, a spiritual component, often called “mind-stuff.” In his conception, the spiritual and material components of the mind can interact with each other. A few researchers such as Eccles and Popper (see [34]) take this position now. If thinking, consciousness, and intelligence require a spiritual component, then it may be difficult or impossible to get a machine to behave much like a human being.

With all this disagreement on what constitutes intelligence, thinking, and understanding, it will be some time before satisfactory definitions are worked out.

1.2 Association

The principle of association may be the most important principle used in intelligence. Briefly put, given that a set of ideas is present in the mind, this set will cause some new idea to come to mind, an idea that has been associated with the set of ideas in the past. This most important principle has been known for hundreds or even thousands of years, but perhaps the best early detailed description was given by a famous 19th century philosopher/psychologist, William James, in his two-volume series, *The Principles of Psychology* [72] (and the abridged one volume version, *Psychology* [73]). In effect, James and other psychologists and philosophers had a very high-level solution to the AI problem by the 19th century, however, they were unfortunate in that there were no computers available at the time with which they could make their ideas more concrete.

In the excerpt below, James gives the principles of association:

When two elementary brain processes have been active at the same time, or in immediate succession, one of them, upon reawakening, tends to propagate its excitement into the other.

But, as a matter of fact, every elementary process has unavoidably found itself at different times excited in conjunction with *many* other processes. Which of these others it shall now awaken becomes a problem. Shall *b* or *c* be aroused by the present *a*? To answer this, we must make a further postulate, based on the fact of *tension* in nerve tissue, and on the fact of summation of excitements, each incomplete or latent in itself, into an open resultant. The process *b*, rather than *c*, will awake, if in addition to the vibrating tract *a* some other tract *d* is in a state of subexcitement, and formerly was excited with *b* alone and not with *a*.

Instead of thinking of summing up “tension” in nerve tissue, today we would think of summing up voltages or currents.

These principles can be better understood with Figure 1.1. If we activate the idea, *a*, and its only associations in the past are with the idea *b* with an activation strength of 0.25 and with *c* with an activation strength of 0.40, then *c* must come to mind. Think, as James does, as if “tension” or electrical current is flowing from *a* into both *b* and *c*. This is shown in Figure 1.1(a). On the other hand, if at some point in time, *b* had been associated with *a* and *d*, and if *a* and *d* come to life, the idea *c* must come to mind, as the sum of currents flowing into it is the greatest. This is shown in Figure 1.1(b). We will often speak of ideas “lighting up” or being “lit.” This is in accord with conventional terminology where people often say that ideas “light up” in their mind. We will also talk about the “brightest” (highest rated) idea as the one that comes to mind.

Some examples of this summation process are worth looking at. One example from James is what occurs when the old poem, “Locksley Hall,” is memorized. Two different lines of the poem are as follows:

I, the heir of all *the ages* in the foremost files of time,

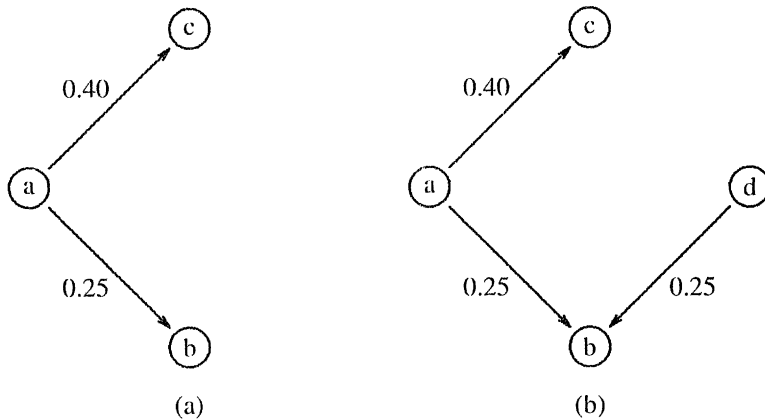


Figure 1.1: Summation of excitements.

and

For I doubt not through *the ages* one increasing purpose runs.

We focus in on the words, “the ages.” If a person had memorized this poem and started reciting the first line, and got to the phrase, “the ages,” why should he continue with the words, “in the foremost files of time,” rather than “one increasing purpose runs?” The answer is simple. While “the ages” points to, or suggests, *both* “in the foremost files of time” and “one increasing purpose runs,” there are those words *before* “the ages” that also, but to a smaller extent, point to “in the foremost files of time.” The summation of the “the ages” with “I, the heir of all” produces a larger value for “in the foremost files of time” than for “one increasing purpose runs.”

A second example from James is the following:

The writer of these pages has every year to learn the names of a large number of students who sit in alphabetical order in a lecture-room. He finally learns to call them by name, as they sit in their accustomed places. On meeting one in the street, however, early in the year, the face hardly ever recalls the name, but it may recall the place of its owner in the lecture-room, his neighbors’ faces, and consequently his general alphabetical position: and then, usually as the common associate of all these combined data, the student’s name surges up in his mind.

The principles of association also form the basis for most TV game shows. The principles have been seen there in their purest form in the shows *Password*, *Password Plus*, and *Super Password*. In the simplest version, *Password*, there are two teams of two players each. One player on each team is given a secret word, short phrase, or name and the object of the game is for this person to say a word that will induce the player’s teammate to say the secret word, phrase, or name. Whichever team gets the right answer scores some points. For example, in one game the secret name was “Jesse James.” The first clue given in the game was “western” but the other person’s response was “John Wayne.” This is fairly reasonable since in many people’s minds, John Wayne is very closely associated

with Westerns. Some other reasonable responses might be “cowboys,” “Indians,” or “eastern.” Since the response was wrong, the other team gets a chance and this time the clue was “train.” Adding together the clues “western” and “train,” some reasonable responses might be “Santa Fe,” “Union Pacific,” or “Central Pacific,” all famous western train companies, but again the contestant got the wrong answer. Finally, after the clues, “frank,” “brother,” and “robber” were given, a contestant got the right answer, “Jesse James,” a famous train robber in the Old West who had a brother named Frank. The game is summarized in Figure 1.2.

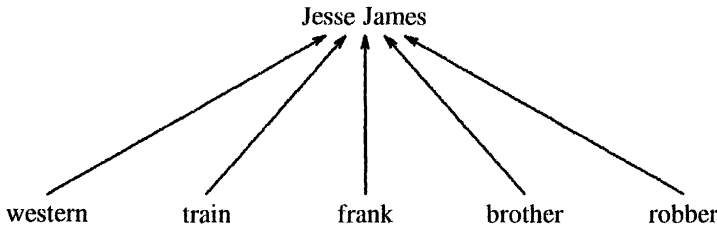


Figure 1.2: A simple game of *Password*.

It is easy to model the process of combining ideas in the following manner. Suppose we assign numeric values to the strength of the associations between ideas. Suppose the associations with “western” are:

| | |
|-----------------|------|
| John Wayne | 0.40 |
| cowboys | 0.30 |
| Indians | 0.25 |
| Santa Fe | 0.25 |
| Jesse James | 0.15 |
| Frank James | 0.10 |
| Union Pacific | 0.05 |
| Central Pacific | 0.05 |

and the associations for “train” are:

| | |
|-----------------|------|
| Amtrak | 0.30 |
| electric | 0.25 |
| Santa Fe | 0.25 |
| Union Pacific | 0.25 |
| Central Pacific | 0.25 |
| Jesse James | 0.15 |

If you want to combine the effects of two different clues, like “western” and “train,” one simple solution is to simply add up how much is being contributed to each of the other ideas in the lists of ideas. In Figure 1.3 we show how “western” and “train” combine to activate all the ideas with which they have been associated in the past. The numbers to the right in Figure 1.3 show the summations. For instance, “western” contributes 0.25 to “Santa Fe”

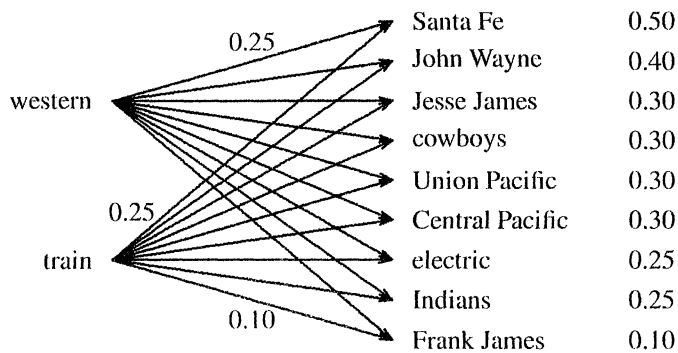


Figure 1.3: How clues in *Password* can combine to produce possible answers. Only a few of the association strengths are shown.

and “train” also contributes 0.25 to “Santa Fe.” When it then comes to guessing an answer, the idea with the highest rating is “Santa Fe.”

For a final example of these principles we now look at their actual use in a simple AI program. Walker and Amsler [245, 246] created a program called FORCE4 whose purpose is to look at newspaper stories and figure out roughly what the story is about. The program does not acquire any kind of detailed understanding of what the article is about, but only tries to put it in a general category, such as weather, law, politics, manufacturing, and so forth. It makes use of a set of subject codes assigned to specialized word senses in the Longman Dictionary of Contemporary English. For instance, the word “heavy” is often associated with food (coded as FO), meteorology (ML), and theater (TH). “Rain-fall” is associated with meteorology (ML). “High” is associated with motor vehicles (AU), drugs and drug experiences (DGXX), food (FO), meteorology (ML), religion (RLXX), and sounds (SN). “Wind” suggests hunting (HFZH), physiology (MDZP), meteorology (ML), music (MU), and nautical (NA).

One story given to FORCE4 was the following:

Heavy rainfall and high winds clobbered the California coast early today, while a storm system in the Southeast dampened the Atlantic Seaboard from Florida to Virginia.

Travelers’ advisories warned of snow in California’s northern mountains and northwestern Nevada. Rain and snow fell in the Dakotas, northern Minnesota and Upper Michigan.

Skies were cloudy from Tennessee through the Ohio Valley into New England, but generally clear from Texas into the mid-Mississippi Valley.

For each important word, the program counts one point for each of its associated ideas.

When you apply this procedure to the above story, you get the following counts:

| | |
|----|-----------------------------------|
| 10 | ML (Meteorology) |
| 4 | GOZG (Geographical terms) |
| 4 | DGXX (Drugs and drug experiences) |
| 3 | NA (Nautical) |
| 2 | MI (Military) |
| 2 | FO (Food) |
| 2 | GO (Geography) |
| 1 | TH (Theatre) |

The results show that it is a weather-related story. Walker [245] reports that when over 100 news stories were submitted to the program,

The results were remarkably good: FORCE4 works well over a variety of subjects—law, military, sports, radio and television—and several different formats—text, tables and even recipes.

1.3 Neural Networking

When scientists became aware that nerve cells pass around electrical pulses, most of them assumed that this activity was used for thinking. Relatively little is known about how networks of nerve cells operate, and determining how networks of nerve cells operate is a major part of the field of neural networking. The second major part of neural networking research centers on the study of computer models of simplified nerve cells. In this book we will deal almost exclusively with the computer-based models.

1.3.1 Artificial Neural Networks

Artificial neural networks represent a way of organizing a large number of simple calculations so that they can be executed in parallel. The calculations are performed by relatively simple processors typically called nodes, artificial neurons, or just neurons. Artificial neurons have a number of input connections and a number of output connections. The input connections serve to activate, or excite, or we may say, “light up” a neuron or they might also try to turn off or inhibit a neuron. An excited neuron then passes this excitement on to other neurons through its output connections. Figures 1.2 and 1.3 can be regarded as diagrams of neural networks. In Figure 1.3 there are the input nodes, “western” and “train,” and the outputs are “Santa Fe,” “John Wayne,” and so forth. The connections between inputs and outputs are called *weights* in neural networking terminology and the value of a weight is what we call the “strength of association.”

A simple artificial neuron is shown in Figure 1.4. For artificial neurons, each connection has associated with it a real value called a weight and each neuron has an activation value. The typical algorithm for activating an artificial neuron, j , given a set of input neurons, subscripted by i , and the set of weights, w_{ij} , works as follows. First, find the quantity, net_j , the total input to neuron j by the following formula:

$$net_j = \sum_i w_{ij} o_i$$

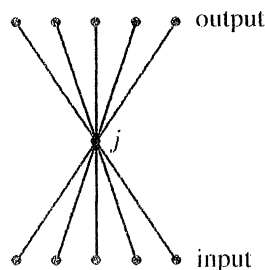


Figure 1.4: A simple artificial neuron, j , with inputs from one set of neurons and outputs to another set.

When the activation value of neuron i , o_i , times the weight w_{ij} is positive, then unit i serves to activate unit j . On the other hand, when the value of unit i times the weight w_{ij} is negative, the unit i serves to inhibit unit j . The activation value of neuron, j , is given by some function, $f(ncl_j)$. The function, f , may be called an *activation function*, *transfer function*, or *squashing function*. One simple activation function is simply to let the sum of the inputs, ncl_j , be the activation value of the neuron. This is how the *Password* network in the last section worked. A second common activation function is to test if ncl_j is greater than some threshold (minimum) value, and if it is, the neuron turns on (usually with an activation value of +1), otherwise it stays off (usually with an activation value of 0). The neural network in Figure 1.5 computes the exclusive-or function and it uses the activation function, $1/(1 + e^{-ncl_j})$. While this function only reaches 0 and 1 at $-\infty$ and ∞ respectively, when the outputs are close enough to 0 or 1 they are counted as being the same as 0 or 1. There are many other possible activation functions that can be used.

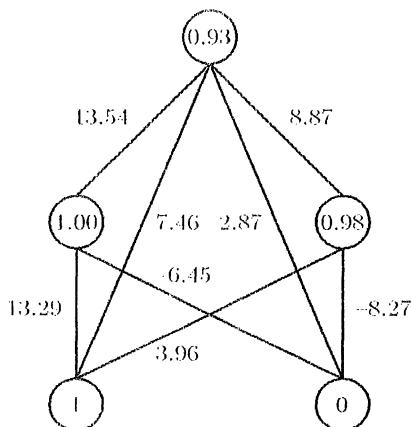


Figure 1.5: This simple network computes the exclusive-or function. The two inputs are made on the bottom layer and the top layer has the answer to within 0.1 of the desired value. In this case the input values are 1 and 0 and the output is 0.93.

Usually the neurons in artificial neural systems have the units arranged in layers as shown in Figures 1.4 and 1.5. These networks have the input layer at the bottom, a *hidden layer* in the middle, and an output layer at the top. The hidden layer gets its name from the

fact that if you view the network as a black box with inputs and outputs that you can monitor, you cannot see the hidden inner workings. When the flow of activation or inhibition goes from the input up to higher-level layers only, the network is said to be a *feed-forward* network. Most often the connections between units are between units in adjacent layers, but it is also possible to have connections between nonadjacent layers and connections within each layer. If there are connections that allow the activation or inhibition to spread down to earlier layers, the network is said to be *recurrent*.

Learning in artificial neural systems is accomplished by modifying the values of the weights connecting the neurons and sometimes by adding extra neurons and weights. There have been a number of learning algorithms that have been proposed and tested for neural networks but the most powerful and most generally useful algorithm is the back-propagation algorithm described in Chapter 3.

Currently, neural networks can be used for many pattern recognition applications, such as recognizing letters, rating loan applications, choosing moves to make in a game, and they can even do simple language processing tasks. One system has been used to automatically drive a van along interstate highways. So far at least, networks are not very well suited to doing complex symbol processing tasks like arithmetic, algebra, understanding natural language, or any task that requires more than a single step of pattern recognition. To produce a system capable of doing much of what human beings can do will require at the very least more complex models and a multitude of different subsystems, each tuned to perform slightly different tasks and all working together.

1.3.2 Biological Neural Networks

As mentioned above, relatively little is known about how biological neural networks operate. For quite a long time it has been assumed that the neurons in the human brain act like the artificial neurons in that they pass around electrical signals, but whereas an artificial neuron receives a single real-value input from each of its input neurons, the biological neurons pass around simple pulses, pulses that are either present or not present. The number of pulses per second going along a connection is an indication of the weight of the connection—more pulses mean a higher weight, fewer pulses indicate a lower weight. In this theory each neuron acts like a little switch, when enough pulses are input a neuron outputs a pulse. The estimates are that there are around 100 billion neurons in the brain with about 1000 connections per neuron and each neuron switches about 100 times per second. This gives a processing rate of around 10^{16} bits per second.² But biological neurons are more complicated than simple switches since they are influenced by chemicals within the cell. One recent discovery is that at least some cells involved in vision are not just sending out plain pulses but in fact are passing around coded messages.³ The shape of the pulse codes the message.

Theories by Hameroff et al. [57] and [133] have each cell acting as a small computer rather than as just a simple switch. The computing would be done in microtubules that make up the cell's cytoskeleton. In this case they estimate a single neuron is processing

²This estimate is taken from the article by Hameroff et al. [57] which in turn was taken from [127].

³A simple description can be found in "A New View of Vision" by Christopher Vaughan, *Science News*, Volume 134, July 23, 1988, pages 58–60. There are many other articles on this topic. See [171] for a list of other references.

about 10^{13} bits per second and the whole brain would be processing at least about 10^{23} bits per second assuming there is some redundancy.

From time to time people have made estimates of how many bits of information the human brain can store based on certain assumptions, but since it is most certainly not known how information is stored and processed in the brain, none of these estimates can be taken too seriously.

In short, little is known about what is really going on in the human brain but new research may soon shed a lot of light on what is going on. Whatever is happening, it is much more complicated than the processing done in the current set of artificial neural network models.

1.4 Symbol Processing

For most of the history of artificial intelligence the symbol processing approach has been the most important one. There are several reasons why symbol processing has been the dominant approach to the subject. First, there were a number of highly impressive symbol processing programs done in the early 1960s. Two of these early systems are described in Chapter 8, the SAINT program of Slagle that could do symbolic integration and the Geometry Theorem Proving system of Gelernter and others. In addition, it seemed obvious to process natural language this way since language consists of symbols. The second reason symbol processing has been dominant is that it seemed as if it would be a very long time before artificial neural networks could be designed that could do such impressive things.

The advocates of the symbol processing approach to AI have proposed the Physical Symbol System Hypothesis (PSSH) (see [138], [139], and [40]). It states that symbols, structures of symbols, and rules for combining and manipulating symbols and structures of symbols are the necessary *and sufficient* criteria for creating intelligence. This means that these features and only these features are required for producing intelligent behavior. Advocates of PSSH assume that the human brain is doing nothing more than manipulations of collections of symbols. In current computers the manipulations are done sequentially, but advocates of this position assume that human minds actually do parallel processing of symbols. It is the *Physical* Symbol System Hypothesis because advocates assume that there are physical states in the brain that correspond to the kind of structures that a symbol processing computer program uses. PSSH advocates also assume that although neural hardware implements the symbol processing abilities of the brain, this hardware is too low level to have to worry about. So, just as Pascal programmers do not have to worry about integrated circuits, symbol processing can concern itself with symbols and structures of symbols without worrying about the underlying neural hardware. Of course, symbol processing adherents acknowledge that neural networking *is* important for lower-level tasks like vision and movement.

The techniques used in symbol processing are very similar to those used in programming in conventional languages such as Pascal and Fortran, however symbol processing emphasizes list processing and recursion and symbol processing methods use symbols rather than numbers. Because in the beginning almost all AI was done in symbol processing languages, some people have defined artificial intelligence as symbolic computing. The most important computer language for AI programs has been Lisp (for list processing

language) and a newer language is Prolog (for programming in logic). For the most part we will use Prolog as a notation for some symbol processing algorithms later in the book because Prolog has some built-in pattern recognition capabilities that Lisp does not have.

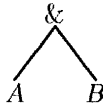
Symbols are defined as unique marks on a piece of paper and in a computer each symbol is represented by a different integer. Two symbols can be equal or not equal, but there are no other relations defined between them. Notice then, that even though symbols are implemented as integers in computer programs, symbols are simpler than the integers that represent the symbols. In addition to being used individually, symbols can also be combined into structures of symbols such as lists or trees. One example of this might be the expression:

$$A \& B.$$

Inside a computer we might find this as a linked list:

$$A \rightarrow \& \rightarrow B \rightarrow nil$$

or as a tree:



Another part of the symbol processing approach is the assumption that there are rules which specify how symbols and structures of symbols are manipulated. Logic and arithmetic provide perfect examples of symbols and how they can be manipulated and combined using rules. Take for example the logic expression, $A \& B \& A$. Now within logic, there is a rewrite rule that says that this expression can be rewritten as $A \& A \& B$. Moreover, there is a rule that says that $A \& A$ can be rewritten as just A . Some rules from arithmetic for manipulating symbols and expressions are: x/x can be replaced by 1, $1 * x$ can be replaced by x , and $x + (-x)$ can be replaced by 0.

The use of rules in symbol processing methods is a key element of the symbol processing approach because everyone who studies human behavior agrees that people exhibit rulelike behavior. For an example of rulelike behavior consider the following case. Suppose a child learns the meaning of a sentence like:

The cat is on the mat.

The child, knowing what a cat is and what a mat is, and what a cat on a mat is, seems to deduce some rules (or form a theory) about how sentences are constructed. Thus, the child can apply the rules and come up with statements like the following:

The dog is on the mat.

The boy is on the mat.

The block is on the mat.

The cat is on the floor.

Some other rules that people form would be “if something is a bird then it can fly” or “if you drop something it will fall.” Such facts are typically coded in a rule format something like this:

if bird(X) then flies(X)

if drop(X) then fall(X)

where X stands for the something. For another example concerning language processing, some researchers have studied how children learn to construct the past tenses of verbs and they have come to the conclusion that the errors that children make show that they are producing rules.

From examples like the above and others, traditional AI researchers have concluded that people have some kind of unconscious machinery that deduces rules as well as some kind of symbol processing architecture that applies them.

Notice, though, that rules are little neural networks where the input and output units have symbolic labels as in this rule:

if a and b then c .

The corresponding network is shown in Figure 1.6. It is a two layer network with inputs labeled a and b and the output unit labeled c . Let the two weights be 1 and let the threshold for unit c be 1.5. Now if unit a and b are both on, ($= 1$), net_c will be 2. Since net_c is greater than 1.5, unit c turns on, otherwise it stays off, ($= 0$). And so it turns out that a key element of symbol processing can be regarded as a form of neural networking.

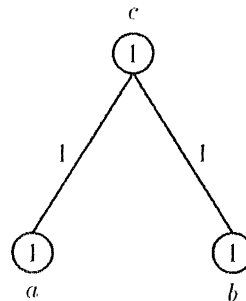


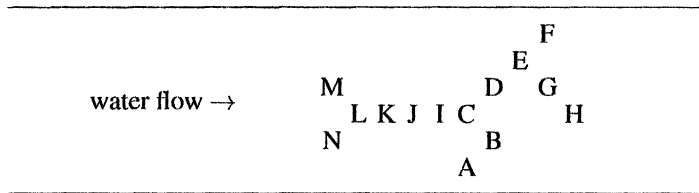
Figure 1.6: The rule, if a and b then c , can be regarded as a neural network where the units a , b , and c can take on the values 0 or 1, the two weights are +1, and the threshold for unit c is 1.5.

Symbol processing techniques have been somewhat successful at doing a number of very narrow but useful tasks involving reasoning and processing natural language.

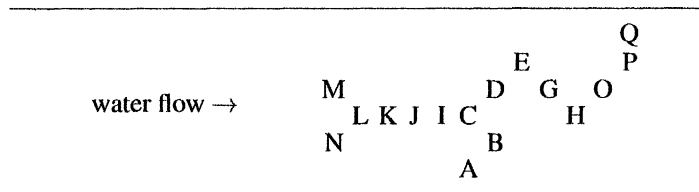
1.5 Heuristic Search

When people encounter a problem they typically have to do some trial and error work on the problem to find the solution. People look at some of the most likely possible solutions *to the problem*, not every possible solution. However, a simple computer program is dumb in that it does not have any way of evaluating the possible solutions to determine which are

the likely ones. This type of program must do an *exhaustive search* of all the possibilities. Very early on it was recognized that for computer programs to solve problems as human beings do, the programs must be able to look at only the likely possibilities. Programs that use some method to evaluate the possibilities are said to do *heuristic search*.



a) a stream where the heuristic succeeds



b) a stream where the heuristic fails

Figure 1.7: In trying to cross a stream in a heavy fog by stepping on rocks, one heuristic is to keep trying to move forward and never back up. Rocks are indicated by letters. In a) the heuristic succeeds but in b) never backing up will produce a failure.

As an example, suppose you were hiking through the wilderness and you came upon a small stream that you needed to cross. Suppose there are small flat rocks in the river and you can step from one rock to an adjacent rock. For example, there is the situation shown in Figure 1.7(a) where each rock is indicated by a letter. Suppose you are on the river bank at the bottom and you want to get to the river bank at the top. A human being will “eyeball” the situation and have it solved in a second. The best (and only) path is from A to B to C to D to E to F. How would you have a computer look at the same situation and find that path? To find one computer solution we could make the problem harder. Suppose you come upon this place in the river, but there is fog and the fog is so thick that you can only see one rock ahead of you. You are clearly going to have to start making guesses as to which steps to take. You will realize that *probably* the best thing to do is to keep going forward as much as possible. What sense would it make to go back in the direction that you came from, or up or down the river? You could tell which way was forward by noticing that the river flows from left to right, so when you make a move, you should try to keep upstream on your left and downstream on your right. Therefore, when you get to rock C, the best thing to do is to go on to D and not to I. Again, when you get to rock E, you will go on to F, rather

than back up by going to G. A computer program could use the same strategy for finding a path across the river and it would find a path as easily as a person lost in a fog. Both you and the computer were doing a heuristic search of a tree, looking for a goal node. If you did an ordinary search of the tree rather than a heuristic search of the tree, you would find a path across, but probably not nearly as quickly. In an ordinary exhaustive search of the tree, when you get to C, you could try going to I. Follow that path and you could go to N. When you got there, you would fail, but you would back up to try M. When that failed, you would back up and go to C, and so on. The heuristic search is intended to get you across the stream as quickly as possible, but there is a possible problem with this method. If you decide that you *must always* go forward and *never* back up, then there will be some locations where your search will fail because there will not always be such a path available (see Figure 1.7(b)). This illustrates another property of heuristic search: while a heuristic search is usually the fastest way to find an answer, you are not always guaranteed that an answer will be found. Of course, when people get a problem to solve there is no guarantee that they will be able to solve it either. An exhaustive search will get the answer but it may take much longer and sometimes so much longer that the search effectively fails.

1.6 The Problems with AI

The results of decades of experimentation with symbol processing, with and without heuristic search methods, has shown that with these methods computers can do *some* tasks that in people would be regarded as intelligent, such as prove theorems, manipulate mathematical formulas, and understand small amounts of natural language. According to some researchers' definitions of intelligence, these programs display intelligence. On the other hand, such programs typically do not learn from their activities and since learning is a key factor in intelligence critics do not see any intelligence in such programs. Then too, the level of understanding that these programs have is severely limited. For example, if you give a program a statement like "John ate up the street," the program might easily conclude that John was eating asphalt. Or, given that "John was in the 100-meter butterfly," a program might think that John was inside a large insect rather than in a swimming event. People say that such programs that do not have the common sense knowledge that people have are *brittle*. In response to this criticism, most symbol processing researchers say they believe their basic methods are valid and the problems can be eliminated by just producing much larger systems. At the moment a very ambitious project known as CYC (see [99]) is attempting to produce a program with a very large number of facts and rules about the world that hopefully will not be brittle. The early estimate was that the program would need about a million rules. As of 1993,⁴ the program had two million and work is continuing at the present time.

1.7 The New Proposals

So while AI has problems, some AI researchers remain optimistic about symbol processing methods but other AI researchers are not and they have started looking into a variety of

⁴ *Computerworld*, May 10, 1993, pages 104–105.

new proposals. Most of these new proposals come to mind quite easily by just denying the elements of the Physical Symbol System Hypothesis. These ideas are that thinking and intelligence require the use of real numbers, not just symbols; that people use images or pictures, not just structures of symbols; and that they use specific cases or memories, not just rules. In addition, there is another proposal that human thinking involves quantum mechanics and quantum mechanics adds extra capabilities that ordinary computing, not even analog computing, can account for.

1.7.1 Real Numbers

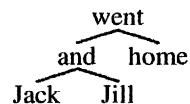
Symbol processing works only with symbols and the only relation defined between symbols is equality, two symbols are equal or not equal. This can work fairly well when the answer to every question is a nice true or false, but in many situations in the real world judgments are fuzzy. The accused person on trial must be proved guilty beyond a reasonable doubt. Music produced by one composer sounds better than the music from another composer. One crime will be judged as more heinous than another. New cars normally look better than older cars. Some government projects are judged as more worthwhile than others. Some chess moves are better than others. It is only natural that researchers think that such judgments are made using some kind of analog computation rather than the simple true/false logic found in symbol processing.

There are a couple of ways to include this analog concept in AI theory, but the most important of these is found in neural networking when the activation values of the units and the weights take on real values. Even though neural networking was present at the beginning of AI research, it was quickly abandoned in favor of symbol processing methods and so it has hardly been investigated until recently. When neural networking methods are applied to AI-type problems it is called *connectionist AI*. One variation on connectionist AI is *parallel distributed processing*, or PDP for short. It uses a specific type of coding within networks.

These methods are fairly good at doing a single step of pattern recognition, but they present connectionist AI researchers with quite a problem as to how to store complicated facts because unlike symbol processing AI where you can use a tree structure to store a fact, in a neural network the facts must be represented as a vector or matrix of real numbers. So for example, if you needed to store away the fact that:

Jack and Jill went home,

it is very straightforward in a digital computer to produce one kind or another of tree structure to represent this such as:



So far there is no established good way to represent this tree structure as a vector or matrix of real numbers, although there are some proposals along these lines.

One position on neural networking is that networks do have some features that are required for intelligence, thinking and reasoning, but conventional symbol processing is

also necessary. In this case the Physical Symbol System Hypothesis is wrong at the point where it says that symbol processing is *sufficient*. One proposal is that the mind may be basically a connectionist computer architecture, but it simulates a symbol processing architecture to do those tasks that are most suited for symbol processing, while still using connectionist methods for other types of problems.

A more extreme position is that neural networking is necessary and *sufficient* and the only reason that symbol processing methods are somewhat successful is that they just approximate what is happening in the mind. To get better performance, connectionist methods are needed.

1.7.2 Picture Processing

MacLennan [102, 103]⁵ has argued that the important features of connectionist AI are the use of real numbers, that the large number of neurons in the brain and eye can be treated mathematically the same as fields (as in magnetic, electric, and gravitational) in physics (see [104]) and that image processing or picture processing is going on in the human mind.

For an example of this suppose we are watching the movie “Jack and Jill’s Greatest Adventure,” with that familiar story:

Jack and Jill went up the hill to fetch a pail of water. Jack fell down and broke his crown and Jill came tumbling after him.

Just watching the movie gives you *images* that are stored away and it is rather difficult to argue that people store these images as some kind of symbolic tree-type representation. Then too, just reading the words will develop images in your mind. Moreover, as Jack starts to fall down you have to predict based on the images that Jack might suffer some damage that will require medical attention, so just working from the images you can do some reasoning. Why use symbols, structures of symbols, and formal rules to do this when picture-based processing will work?

The fact that people do store many memories as pictures and do at least some of their reasoning about the world using pictures ought to be one of the most obvious principles of all, yet it has been neglected, in part due to the predominance of symbol processing and in part due to the fact that processing pictures is hard compared to processing symbols. Unfortunately, at this point in time image processing is still fairly underdeveloped and has not been used in conjunction with representing the real world in programs where the goal of the program is to reason about the real world. Of course, simple image processing has been used by robots and in programs to recognize patterns such as handwritten or typed digits and letters of the alphabet.

1.7.3 Memories

The final key feature of the Physical Symbol System Hypothesis that can be criticized is the idea that people take in large amounts of experience from the real world, condense all these specific instances down to a handful of rules, and then people work from these rules to

⁵In addition to discussing how connectionist programs might store knowledge about the real world, MacLennan also reviews the symbol processing position in these papers so they are quite worthwhile and they are online.

solve new problems. An example of this is that when you drop something, call it, X, where X may be a rock, a piece of paper, or a feather. If you have done a lot of experimenting with dropping various things, then you will derive the rule: *if you are holding something and you let go, it will fall straight down*. In a symbol processing representation you are likely to code this as something like:

if letgo(X) then fall(X).

Yet there is a problem with such a rule because it only applies under certain conditions. If the air is moving and X is a feather or a flat piece of paper, then it will not fall straight down and it may remain in the air for quite some time before reaching the ground. But, if a piece of paper is crumpled up it will fall faster than if it is flat. If the air is moving very rapidly, even a rock will not fall straight down. Then what about the case we have all seen on TV where an astronaut on board a spaceship in orbit around the Earth lets go of something and rather than falling⁶ it simply floats in midair? So a humanly coded set of rules is subject to the same problem that comes up in conventional computer programming where you must consider every possible permutation of the input data. Thus a rule-based program where the programmer has neglected to take into account wind velocity could conclude that if someone dropped a feather in a tornado the feather will fall straight to the ground, another example of the brittleness of conventional programs. So far, generating rules from data has not worked especially well either except for very small problem domains, domains much smaller than the real world domain.

One way to eliminate the problems involved with finding and using rules is to just not bother with the rules. If you have done your experiments of dropping various things under various conditions and seen the experiments done in space, then when someone asks you what will happen if you drop something all you have to do is reference your memories of your experiments to get the answer. This idea that people use simple memories to solve many ordinary real world problems is now getting a lot of attention although it is being done in the context of symbol-based methods, not in a picture-based context. These methods are called *case-based* and *memory-based*.

1.7.4 Quantum Mechanics

The possible application of quantum mechanics to thinking comes up in a number of ways. As already mentioned, Penrose in his two well-known popular books [148, 149] argues that consciousness is necessary for intelligence and quantum mechanics is responsible for consciousness, and moreover, that QM contains a nonalgorithmic component that cannot be duplicated by digital computers. Second, Vitiello [240] has proposed a quantum mechanical memory system that has the useful property that no matter how many memories this system has stored, one more can always be added without damaging any of the old memories, so in effect you get an unlimited memory. Finally, there is the idea that quantum mechanics might allow faster than light communication and this would explain the persistent reports of mind reading and predicting the future. For an argument in favor of this see [79]. Nanopoulos [133] has a quantum mechanical theory of brain function that

⁶The physics people will explain it by saying the object, the person, and the spaceship are really all falling at the same rate.

fits the psychological theories of William James and Sigmund Freud. Unfortunately, the application of quantum mechanics to thinking is still in a very early stage of development, it is more of a hope than any sort of concrete, testable proposal.

1.8 The Organization of the Book

The book starts with some of the lowest-level vision problems in Chapter 2 and then, generally speaking, the book goes on to cover higher and higher level problems until this progression ends with natural language processing in Chapter 10. The principles found at the beginning in vision systems can be found in slightly different forms all the way up to the highest levels. First, Chapters 2 and 3 illustrate the most important and useful pattern recognition and neural networking methods. Chapters 4 and 5 then give the approximate symbolic equivalents to the material in Chapters 2 and 3. The theme of Chapter 6 is that the methods presented so far are much too simple to produce programs with humanlike behavior. What is really needed is a much more complex architecture *and* a method for storing and retrieving knowledge in that architecture. Chapter 7 is to some extent an extension of the knowledge storage and retrieval problem in that storing, retrieving, and using cases rather than the traditional classical method of using rules is the theme. To a large extent Chapters 8, 9, and 10 are examples and applications of the principles given in Chapters 1 through 7, although, of course, Chapters 8 and 9 also develop the heuristic search theme as well.

One of the key ideas in this book is, of course, that the new methods need to be studied and worked on in order to get programs to achieve human levels of performance in dealing with problems, especially in dealing with the whole range of real world problems. However, all these methods, the symbolic and the neural and the memory-based all represent different ways of doing *pattern recognition*. Pattern recognition can be defined as the ability to classify patterns like the letters of the alphabet, other written symbols, or objects of various sorts; however, pattern recognition can also be used to try and find the more abstract and hidden patterns that exist within economic data or social behavior. Pattern recognition can also be used to describe the process of finding patterns that are close to each other in situations where the goal is not to do formal classification. Pattern recognition is also a formal academic field of study, typically found in electrical engineering or computer science departments, where the goal is once again to recognize patterns, either the visual ones or the more abstract ones.

1.9 Exercises

1.1. In the original Turing test [238], two people, a man and a woman go into the Turing test room while a third person asks them questions through a teletype system. Suppose the man is A and the woman is B but the third person knows them as X and Y. The problem for the third person is to try to determine whether X is male and Y is female or if X is female *and* Y is male. In the game, Y will try to help the questioner make the correct identifications but X will try to confuse the questioner. Then Turing says:

We now ask the question, What will happen when a machine takes the part of A in this game? Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and woman? These questions replace our original, "Can machines think?"

Would this version of the Turing test be any better at identifying thinking than the usual version where the game is simply to determine whether the entity in the Turing test room is a person or a computer?

Consider this too: one person posted a note in the Usenet comp.ai.philosophy newsgroup saying that Turing was noted for being a playful man and that maybe this whole Turing test was just a playful joke.⁷

You might actually want to try the human only version of this test in class. One way that is said to be very effective in determining who is male and who is female is to ask X and Y false questions such as "What is a Lipetz-head screwdriver?" Firschein⁸ reports that in his experiments with the test: "Once this false question approach is discovered, few students can successfully fool the class."

1.2. The strong AI position is that certain types of computing are thinking. Suppose this is true. Does this mean that computers will be able to write great music, great poetry, create great art, and so on, or are these things something that only people can do?

1.3. Here are some examples of third and fourth grade arithmetic word problems:

Matt has 5 cents. Karen has 6 cents. How many cents do they have altogether?

Kathy is 29 years old. Her sister Karen is 25 years old. How much older is Kathy than Karen?

If Mary sells 5 pencils at 6 cents each, how many cents will she have altogether?

It is not very hard to get a computer program to do a fair job of reading and solving these problems. These problems are quite simple to program because all children know at this grade level is how to add, subtract, multiply, and divide integers. They do not know about negative numbers or fractions. All they (or a program) have to do is pick off the numbers and then decide which operation to apply. Subtraction must always produce a positive number and division must always produce an integer. Also, the problems are loaded with phrases such as: "have altogether," "how much more," and "at this rate, how many."

Write a program that will look at the words in problems, much as the FORCE4 program did in Section 1.2 and then have it decide on what operation to apply to get the answer. Consult third and fourth grade textbooks for more problems.

Also consider the following alternative strategy. Instead of using individual words alone to choose the operation to apply, try using each *pair* of adjacent words in the problem. For example, "much more" will suggest subtraction and "many altogether" will suggest addition or maybe multiplication. Of course this will produce a longer list of items to store, but see if it produces better results.

⁷From Kenneth Colby, UCLA Computer Science Department, Message-ID: <3k4iub\$P8n@oahu.cs.ucla.edu>, 14 Mar 1995 09:14:51 -0800.

⁸"Letters to the Editor," Oscar Firschein, *AI Magazine*, Fall 1992.

In fact, it is easy to create a fairly small program that can *learn* to do these problems. Give the program some sample problems and let it break the text down into pairs of words. For the first problem above, you get the pairs, “Matt has,” “has 5,” “5 cents,” “cents,” and so forth. Associate addition with each of these pairs. Expose your program to many such problems, and then test it with some of the problems you have trained it on, as well as on some unknown ones, and see how effective it is.

Whether you program this problem or not, you can still evaluate the effectiveness of the techniques that have been suggested as well as suggest more techniques that may work. Consider whether or not you could use these techniques or similar ones to do harder problems like:

John went to the store and decided to buy 4 pieces of candy at 10 cents each.
He gave the clerk 50 cents. How much change should he receive?

1.4. Rather than trying to classify arithmetic word problems you may want to classify Usenet news articles on two or more topics. It is probably best to choose articles from two very different newsgroups. After you train your program on the two classes, give the program some additional articles to see how well it classifies them.

1.5. For the network in Figure 1.5, show that when any two binary digits are given to the two input units the correct value (to within 0.1) of the exclusive-or of the two inputs appears on the output unit. Compute by hand and give the hidden unit values as well.

1.6. If we use a neural network where output units have real values for the threshold values and weights, show the networks corresponding to these two rules:

if a or b then c
if $(a$ or $b)$ and not c then d

1.7. Is the heuristic search suggested for crossing the river on rocks a realistic model of how people would find a path across the river if there was no fog? If it is not, how do people do it?

1.8. For some extra background on AI, read and summarize one or more of the following articles, all found in the Winter, 1988 issue of *Daedalus*:

“One AI or Many?” by Seymour Papert,
“Making a Mind vs. Modeling a Brain” by Stuart and Hubert Dreyfus,
“Natural and Artificial Intelligence” by Robert Sokolowski,
“Much Ado About Nothing” by Hilary Putnam,
“When Philosophers Encounter Artificial Intelligence” by Daniel C. Dennett.

1.9. Stuart and Hubert Dreyfus are two noted critics of artificial intelligence and they give their criticism in the book, *Mind Over Machine* [28]. Read this book and summarize their criticisms and then state whether or not you agree with them and why. (A good due date would be near the end of the course.)