Biological metaphors and the design of modular artificial neural networks

Abstract

In this thesis, a method is proposed with which good modular artificial neural network structures can be found automatically using a computer program. A number of biological metaphors are incorporated in the method. It will be argued that modular artificial neural networks have a better performance than their non-modular counterparts. The human brain can also be seen as a modular neural network, and the proposed search method is based on the natural process that resulted in the brain: Genetic algorithms are used to imitate evolution, and L-systems are used to model the kind of recipes nature uses in biological growth. A small number of experiments have been done to investigate the possibilities of the method. Preliminary results show that the method does find modular networks, and that those networks outperform ‘standard’ solutions. The method looks very promising, although the experiments done were too limited to draw any general conclusions. One drawback is the large amount of computing time needed to evaluate the quality of a population member, and therefore in chapter 9 a number of possible improvements are given on how to increase the speed of the method, as well as a number of suggestions on how to continue from here.

Introduction

‘Your quote could have been here.’ Modularity1 is everywhere. From astrophysics to quantum mechanics, one can see modularity. In biology, the growth and development of living organisms is modular. The human brain also is probably highly modular. Modularity in the brain can be identified at many different levels, physical as well as functional. In this thesis, it will be argued that when modelling the brain using artificial neural networks in order to create so-called intelligent software, exploiting modular design principles will result in better networks. Also, it will be shown that using techniques based on biological genetics and growth to search for optimal neural network topologies can result in an all-round search for good topologies for a variety of problems. Research goals Preliminary results show that using modularity when designing artificial neural networks can improve their performance. As will be set forth in chapter 2, the topology of a network has a large influence on the performance of that network but, so far, no method exists to determine the optimal topology for a given problem because of the high complexity of large networks. The (human) brain is a large scale neural network based on modularity and it might therefore be a good idea to do a reverse engineering of the genetic search and development processes that led to the brain. The genetic search in nature resulted in the usage of a kind of recipes (instead of blueprints) to describe the development of the organism. The goal of this research was to develop a method with which good neural network topologies could be found using a computer. Because of the excellent results found in nature, we strived to keep our method as biological plausible as possible, so the found topologies should have a 1 In this thesis, modularity is defined as a subdivision in identifiable parts, each with its own purpose or function. 2 1 Introduction high degree of modularity and should be found using genetic search and recipes instead of blueprints. L-systems (which can be seen as a kind of recipe) are able to encode repeated patterns (modules) with a complex internal structure efficiently. Genetic algorithms (used to simulate evolution) and L-systems are treated in chapters 3 and 4. Chapter 5 provides a more thorough treatment of modularity in nature and also discusses the usefulness of this principle for our research. The thesis is divided in three parts: in the first part (chapters 2 through 4) introductions are given to the three main disciplines used in this research. The second part (chapters 5 through 7), gives a more thorough treatment of the ideas behind this research and the software developed. Finally, in the last part, (chapters 8 and 9), the results of a number of experiments are reported and recommendations for further research are given. In the remainder of this chapter the used disciplines are briefly introduced. Neural networks Ever since computers were invented, humans have tried to imitate (human) intelligent behaviour with computer programs. This is not an easy task because a computer program must be able to do many different things in order to be called intelligent. Also, the meaning of the word intelligence is somewhat unclear, because many different definitions exist. The methods used to achieve artificial intelligence in the early days of computers, like rule based systems, never achieved the results expected and so far it has not been possible to construct a set of rules that is capable of intelligence. Because reverse engineering proved to be successful in many other areas, researchers have been trying to model the human brain using computers. Although the main components of the brain, neurons, are relatively easy to describe, it is still impossible to make an artificial brain that imitates the human brain in all its detailed complexity. This is because of the large numbers of neurons involved and the huge amount of connections between those neurons. Therefore large simplifications have to be made to keep the needed computing power within realistic bounds. An artificial neural network consists of a number of nodes which are connected to each other. Each of the nodes receives input from other nodes and, using this input, calculates an output which is propagated to other nodes. A number of these nodes are designated as input nodes (and receive no input from other nodes) and a number of them are designated as output nodes (and do not propagate their output to other nodes). The input and output nodes are the means of the network to communicate with the outside world. There are a number of ways to train the network in order to learn a specific problem. With the method used in this research, backpropagation, supervised learning is used to train the network. With supervised learning, so-called input/output pairs are repeatedly presented to the net. Each pair specifies an input value and the output that the network is supposed to produce for that input. To achieve an internal representation that results in the wanted input/output behaviour, the input values are propagated through the nodes. Using the difference between the resulting output and the desired output, an error is calculated for each of the output nodes. Using these error values, the internal connections between the nodes are adjusted. This process is described in detail in chapter 2. Neural networks 3 Genetic algorithms Darwinian evolution, where fit organisms are more likely to stay alive and reproduce than non-fit organisms, is modelled by genetic algorithms. A population of strings is manipulated, where each string can be seen as a chromosome, consisting of a number of genes. These genes are used to code the parameters for a problem for which a solution has to be found. Each string can be assigned a fitness, which indicates how good the string is as solution for the problem. As with natural selection and genetics, where the chance of reproduction for an organism (and thus its genes) depends on its ability to survive (its fitness), the strings used by the algorithm reproduce proportional to their fitness. A new generation is created by selecting and recombining existing strings based on pay-off information (their fitness) using a number of genetic operators. The most commonly used operators are selection, crossover, inversion and mutation. They are described in detail in chapter 3. In that chapter also a more thorough treatment is provided of the basic mechanisms of genetic search as implemented by a genetic algorithm. L-systems The development of living beings is governed by genes. Each living cell contains genetic information (the genotype) which determines the way in which final form of the organism will develop (the phenotype). This genetic information is not a blueprint of that final form, but can be seen as a recipe. This recipe is followed not by the organism as a whole, but by each cell individually. The shape and behaviour of each cell depends on the genes from which information is extracted and this in turn depends on which genes have been read in the past and on influences from the environment of all the neighbouring cells. So the development is solely governed by the local interactions between elements that obey the same global rules. In order to model this development in plants, the biologist Aristid Linden mayer developed a mathematical construct, called L-systems. By using so-called rewriting rules, with an L-system, a string can be rewritten into another string by rewriting all characters in the string in parallel into other characters. The application of a rewriting rule depends on which rules have been applied in the past and on the neighbouring characters of the character to be rewritten. Chapters 4 and 6 describe the standard L-systems and the L-system used during this research. Overview The method that resulted from the combination of the three techniques can be described as follows. When looking for a network structure that is able to learn a task, a genetic algorithm is used to produce production rules for an L-system. These rules result in a network structure. That structure can then be evaluated by looking at the extent in which the network can learn the task. This results in a fitness which is coupled to the original genetic coding of the production rules. The fitness enables the genetic algorithm to evolve a set of production rules that generate an optimal network structure.

Neural Networks

‘The human brain is like an enormous fish. It is flat and slimy and it has gills through which it can see. Should one of these gills fail to open, the messages transmitted by the lungs don’t reach the brain. It is as simple as that.’ — Monty Python on a disease called hooping cough ‘We animals are the most complicated things in the known universe.’ This quote from the biologist Richard Dawkins [DAWK86] (p.1) assents the complexity of the problems seen in biology. The brain is perhaps the most complex organ of an animal, particularly in humans. Discovering how the brain works, and how it is able to be intelligent, is probably the most challenging task ever. The human brain The human brain is a network of a huge number of interconnected neurons. Each neuron has a very complex bio-electrical and bio-chemical behaviour, though its basic computational principles are believed to be very simple: it adds its input and performs a threshold operation. In other words: every neuron is capable of adjusting its output as a relatively simple function of its input. How can it be that such a rather simple basic unit is able to generate such a complex behaviour? The main key to the answer is the cooperation and interaction between neurons. Although they work relatively slow compared to modern computers, there are a lot of them and they all operate in parallel. This combined effort of these large numbers of neurons is what is believed to be the origin of human intelligence. What intelligence means, and what it is, is still unknown. But so far we humans seem to be the only ones who, in the future, may be able to provide the answer1 . There is little doubt that if we discover the origin of intelligence, we will owe it to the wonderful organ in our head. 1 An interesting question is whether the human brain is capable to understand its own functioning. Gödel’s theorem suggests that there may be ‘ideas’ which can not be understood by the brain, assuming the brain can be described as a formal system... (see e.g. [HOFS79] and [PENR89]) 6 2 Neural Networks Artificial intelligence The artificial intelligence community has for a long time been trying to imitate intelligent behaviour with computer programs. This is not an easy task because a computer program must be able to do many different things in order to be called intelligent. Something that has caused a lot of confusion is the definition of artificial intelligence: the Webster’s Dictionary alone, for example, gives 4 definitions of artificial intelligence: 1. An area of study in the field of computer science. Artificial intelligence is concerned with the development of computers able to engage in human-like thought processes such as learning, reasoning, and self-correction. 2. The concept that machines can be improved to assume some capabilities normally thought to be like human intelligence such as learning, adapting, self-correction, etc. 3. The extension of human intelligence through the use of computers, as in times past physical power was extended through the use of mechanical tools. 4. In a restricted sense, the study of techniques to use computers more effectively by improved programming techniques. Allan Turing has proposed a test that should be satisfied in order to speak of artificial intelligence. In this test, known as the Turing Test [TURI63], a person, say Q, is placed in a room with a terminal connected to two other sites. At one of the two terminals a person is situated and at the other a computer. By asking questions, Q must determine at which of the two terminals the computer is situated. Turing is willing to accept the computer program as being intelligent if Q fails. Of course the set-up of the test should make it impossible to decide who is who by measuring response-time etcetera. The Turing Test is disputed because some believe it is possible to deceive Q without having an intelligent program. The traditional ways of designing intelligent systems, like rule-based systems, never achieved the results that were expected at the time people started to realize that computers could be used for more than just calculating numbers. So far it has not been possible to construct a set of rules that is capable of intelligence. There are some expert systems able to compete on a specialist level in very narrow areas, but there is no program yet that is capable of functioning in everyday situations. The problems encountered with the more traditional methods urge more and more researchers to look for other approaches. A principle that has proved to be effective on many other occasions is reverse engineering (looking for something that works, trying to understand it and then rebuild it). In this particular case, it means looking at the natural brain and using its processing principles in a computer program. The neuron The neurons (the nerve cells) in the brain are the cells responsible for our thinking. Because all artificial neural networks are based on the way neurons work, it is important to have an idea of their functioning. The neuron, as shown in figure 1, can be divided in three functional parts: the dendrites, the body and an axon. The dendrites are the information collectors of the neuron. They receive signals from other neurons, and transmit those signals in the form of electrical impulses to the cell body. The body collects all these impulses and if the summed charge of all these impulses exceeds a certain threshold, the cell body activates the axon. The axon transmits this The neuron 7 activation, again as an electrical impulse, to the dendrites of other cells. To enable the brain to Figure 1. The neuron. change its internal processing and learn, the influence of one neuron to another is thought to be variable. The learning takes place at the junction of the axons and dendrites. Each axon splits into a number of these junctions, called synapses. Each synapse responds to the electrical impulse in an axon by releasing certain neurotransmitters. These chemical substances, when reaching the dendrite, cause the electrical activity that is collected at the body of the receiving cell. Changes in this process causes learning and development. Neurons connected: the brain The brain is made up of around 1011 neurons. It should be clear that it is quite impossible to fully connect each neuron with each other neuron. It has been estimated that a full connectivity would result in a head with a 10 kilometre diameter, because of the huge amount of wiring [HEEM91]. In order to reduce the amount of connections, the brain is divided in different modules at several levels. The clearest division of the brain is the division in a right and a left half, which function to a large extent independently. Instead of being fully connected, they are connected by a relatively small amount of connections through a structure called the corpus callosum. At a smaller scale the brain is divided in a number of functional areas, for example the visual area, auditory area, and separated sensory and motor areas, and so on. Between these areas too, only a relatively small number of connections exist. Gazzaniga [GAZZ89] describes a patient who was not able to name the colour of red fruit, after suffering from a head injury. The patient was able to name the colour of every other object presented to him, including other red objects. But when presented with red fruit, the answers where random. Apparently the specific connections between the areas where fruit and the colour red are recognized were lost. This example suggests a strong modularity in the brain, even at a smaller scale than the functional areas. Indeed, a lot of psychological and physiological research indicates a very strong modularization of the brain (see chapter 5). Despite of this highly specific structure of the brain, there still are a lot of connections between neurons. On average every neuron receives input from about 10 thousand synapses. 8 2 Neural Networks Because of these huge numbers of neurons and connections, it is clear why researchers have not been able to do a computer simulation of the brain, for the amount of computation needed is huge. Let us make an estimate of the computing power of the brain. Every axon is able to transmit one pulse every 10 milliseconds, and because the synapses are unable to use the differences in the amplitude of the impulse, the axon can be seen as a cable transmitting 100 bits per second. Combined with the total number of axons, this results in roughly 1013 bits per second. This is an estimation of just the data transmission in the brain: the amount of computation is even more staggering. Jacob Schwartz [SCHW88] estimates the total amount of arithmetic operations needed to simulate the brain in every detail as high as 1018 per second, needing 1016 bytes of memory. This is probably a million times as fast as the fastest supercomputer available in the next decade. It is also a motivation to do a lot of research in massive parallel computers. Artificial neurons As a consequence of the huge complexity of the human brain and the state of current hardware technology, it is impossible to build an artificial brain that imitates the natural brain in all its detail. So in order to make use of its functional principles, we are forced to make (very) large simplifications regarding the computations performed by the neurons and their connectivity. Artificial neurons take their input (real numbers), and act f(stim) stim n i 1 wi xi θ Figure 2. The basic processing element. determine their output as a function of their input. Most of the time the total stimulation of these processing elements is simply the sum of all individual inputs multiplied by their corresponding weights. Sometimes a bias term θ is added in order to shift the sum relative to the origin: . stim n i 1 wi xi θ The analogy with the real neuron is obvious: in the brain the activation of a neuron is transmitted through the axon and arrives at the dendrites of the other cells where the strength of the synapses determines the extent in which the other cells are stimulated. Although real neurons give either excitatory or inhibitory signals at their synapses, in our model the weights of one node can be positive and negative. There is a clear distinction between the stimulation of a processing element and its activation, the latter is usually implemented as a function of the first: act f(stim) f .         n i 1 wi xi θ Sometimes it is implemented as a function of the stimulation and the previous activation1 : act(t) f(act(t 1),stim(t)). 1 Because of the digital simulation of the networks, generally time is considered to be measured in discrete steps. The notation t-1 indicates one timestep prior to time t. Artificial neurons 9 Although f is chosen depending on the kind of network that is being used, the basic functioning of the processing elements is the same in most neural networks, because it is assumed that this kind of artificial neuron implements the basic computational principles of the biologic neuron. Artificial neural networks As is the case with the brain, artificial neural networks are made up of a number of interconnected processing elements. The reader may now wonder what advantages neural networks offer compared to more traditional methods. One of the largest problems with almost all traditional artificial intelligence techniques is that the programmer has to supply all the knowledge (which is very often incomplete, unknown, or wrong). Even if a human expert ‘knows’ how to make certain decisions, for example in the medical world, he is very often incapable of telling exactly why and on what grounds those decisions are made. This is why a new area of research, called knowledge engineering, has arisen which purpose is to find methods for acquiring knowledge. Very often it happens that the knowledge collected cannot be described by definite rules, but has to be described using e.g. statistical reasoning methods and fuzzy logic. This difficulty of finding rules can be solved with neural networks, because they do not have to be told what to do, but are able to learn it by themselves. Neural networks are capable of autonomously discovering regularities, and extract knowledge from examples in complex task domains (like human experts). This is why the neural network approach is so promising, especially in areas lacking ‘absolute knowledge’. The fact is that all traditional methods, like rule based systems, and even newer methods like fuzzy logic can be seen as special cases of neural networks. Since it is possible to construct the logical NAND function using a neural net even proves that everything a computer can do can be done using a neural network, because by combining these NAND networks a complete computer can be built. This shows how powerful artificial neural networks in the future will become, and even now there are already many examples available showing the strength of neural networks, although the neural network research is just at its beginning. Some areas where neural networks are successfully used are (for a more exhaustive overview see e.g. [HECH90]): - handwritten character recognition, - image compression, - noise filtering, - broomstick balancing, - automobile autopilot, - nuclear power-plant control, - loan application scoring, - speech processing, - medical diagnosis, but this list is far from complete, and new applications seem to appear every day. Backpropagation There are a lot of different neural network paradigms, of which backpropagation probably is the best known. It was formalized first by Werbos [WERB74] and later by Parker [PARK85] and 10 2 Neural Networks Rumelhart and McClelland [RUME86]. It is a Figure 3. A typical backpropagation network. multi-layer feedforward network that is trained by supervised learning. A standard backpropagation network consists of 3 layers, an input, an output and a hidden layer. The processing elements of both input and output layer are fully connected with the processing elements of the hidden layer, as shown in figure 31 . The fact that it is feedforward means that there are no recurrent loops in the network. The output of a node never returns at the same node, because cycles are not allowed in the network. In standard backpropagation this can never happen because the input for each processing element always comes from the previous layer (except the input layer, of course). This, again, is a large simplification compared with the real brain because the brain itself appears to contain many recurrent loops. Supervised learning means that the network is repeatedly presented with input/output pairs (I,O) provided by a supervisor, where O is the output the network should produce when presented with input I. These input/output pairs specify the activation patterns of the input and output layer. The network has to find an internal representation that results in the wanted input/output behaviour. To achieve this, backpropagation uses a two-phase propagate-adapt cycle. In the first phase the input is presented to the network and the activation of each of the nodes (processing elements) of the input layer is propagated to the hidden layer, where each node sums its input and propagates its calculated output to the next layer. The nodes in the output layer calculate their activations in the same way as the nodes in the hidden layer. In the second phase, the output of the network is compared with the desired output given by the supervisor and for each output node the error is calculated. Then the error signals are transmitted to the hidden layer where for each node its contribution to the total error is calculated. Based on the error signals received, connection weights are then adapted by each node to cause the network to converge toward a state that allows all the training patterns (input/output pairs) to be encoded. For a more detailed description of backpropagation networks, the reader is referred to appendix A.