Heather Dewey-Hagborg

Nancy Hechinger

Thesis, Spring 2007

April 23, 2007

Creating Creativity

Table of Contents

1.0 Abstract

This thesis discusses my research into creativity as an emergent property of memory and explores the possibility of machine creativity through experimentation with biologically inspired electronic architectures.

1.1 Introduction

One intriguing characteristic of the human mind is its ability to be *creative*, that is the ability to generate an output, an action, or a phrase, that is not explicitly learned; to evolve memory through synthesis and noise, arriving at ideas and solutions that seem to come out of nowhere; that appear to be completely *new*.

Creativity, like consciousness or intelligence, is a fundamentally social and perceptual phenomenon. There is no list of ingredients, secret formula, or divine synthesis that gives rise to any of these qualities. A person, an animal, or a machine, is creative if we deem them to be so, if our culture has shaped us to believe it. Nonetheless, the structure of the biological brain and the process by which it evolved lends it the *capacity* for these qualities and the perception of them.

This thesis is concerned with the possibility of *creating* creativity. It proposes that by constructing electronic structures using processes and materials inspired by biology we can enable a creative capacity in machines.

Through utilization of principal component analysis and neural network techniques I have developed a creative software program. Like a living creature, each

instantiation of the code is unique; though each may experience similar information, they apprehend it and remember it in their own anomalous way. Following from this, they interpret new and ambiguous stimuli divergently. This project examines the core possibility of how it is that human beings come up with ideas which are *new to them*, and how this capability can be translated to the realm of machines.

2.0 Personal Statement

"Art = imitation of nature in her manner of operation"

-John Cage, Themes and Variations

"Artificial Life is the study of man-made systems that exhibit behaviors characteristic of natural living systems. It complements the traditional biological sciences concerned with the analysis of living organisms by attempting to synthesize life-like behaviors within computers and other artificial media. By extending the empirical foundation upon which biology is based beyond the carbon-chain life that has evolved on Earth, Artificial Life can contribute to theoretical biology by locating life-as-we-know-it within the larger picture of life-as-it-could-be."

-Christopher Langton, Artificial Life

There are some questions we ask in childhood which never find resolution. What does it mean to be alive? Why should I be good? Why are people different? What is thinking? Though often laughed off and dismissed when departing the child's lips, these

are among the core questions of philosophy. In many respects they are the most fundamental questions imaginable, yet they remain unanswered and possibly unanswer*able*. The question of thought, in particular, has intrigued me for as long as I can remember.

I have been interested in artificial life since I learned how to program a computer. Right away I started playing with neural networks and genetic algorithms. I built robots and explored ideas of emergence in language and collective behavior. The field immediately appealed to my artistic sensibility. Coming from a background in electronic art it seemed a natural path to explore. Conceptually, it takes John Cage's idea of indeterminacy and chance operations one step further; the work is not only free of the artist's hand, it actually has a life of its own. In this way I am synthesizing the two quotations above to create an art form which abstracts nature's manner of operation as the seed of a new nature.

For the past 5 years I have been working at the intersection of art and artificial life, exploring the philosophical underpinnings of computational media through an artistic lens. The fundamental question for me now is whether it is in fact possible to construct an artificial mind, or whether "mindfulness" is a quality peculiar to human beings and interaction in human society.

3.0 Research

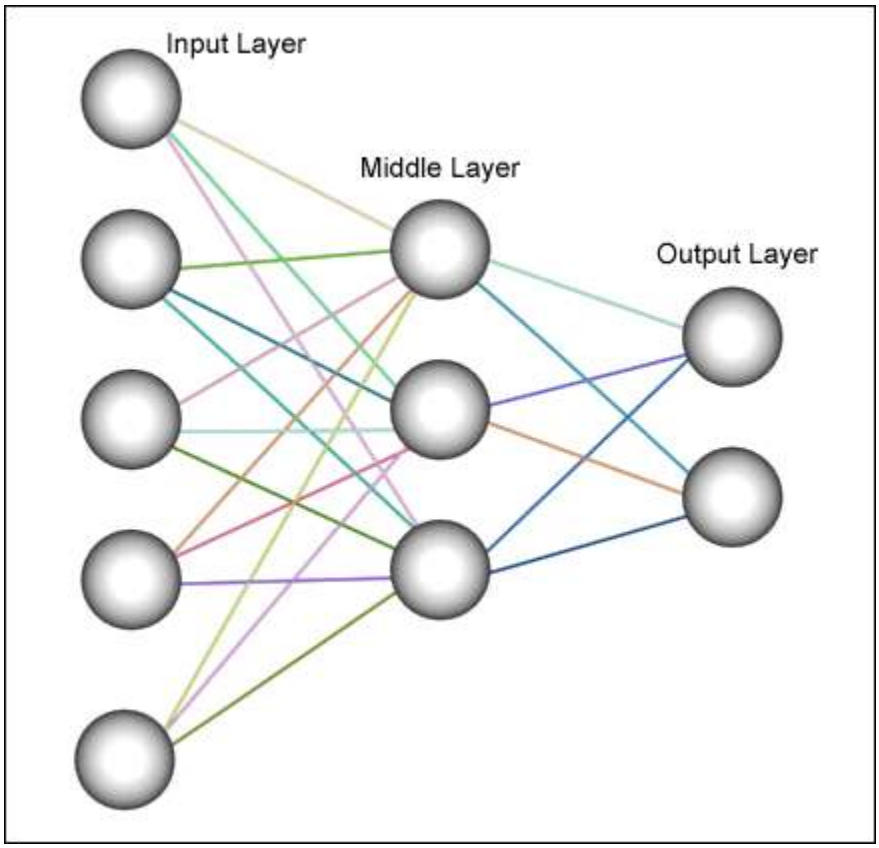
Spurious Memories draws on research from diverse fields including computer science, psychology, philosophy and electrical engineering. It's physical form stems from a combined history of Conceptual, Installation and New Media Art. In the following

sections I will explain the background research underpinning the technical aspects of the project.

3.1 Neural Networks

Neural network research is a sub-section of machine learning concerned with systems of simple interacting components which give rise to intelligent behavior. Numerous types of neural networks have been developed, but in this paper we will be focused on Hebbian learning, attractor networks and Self-Organizing Maps(SOMs).

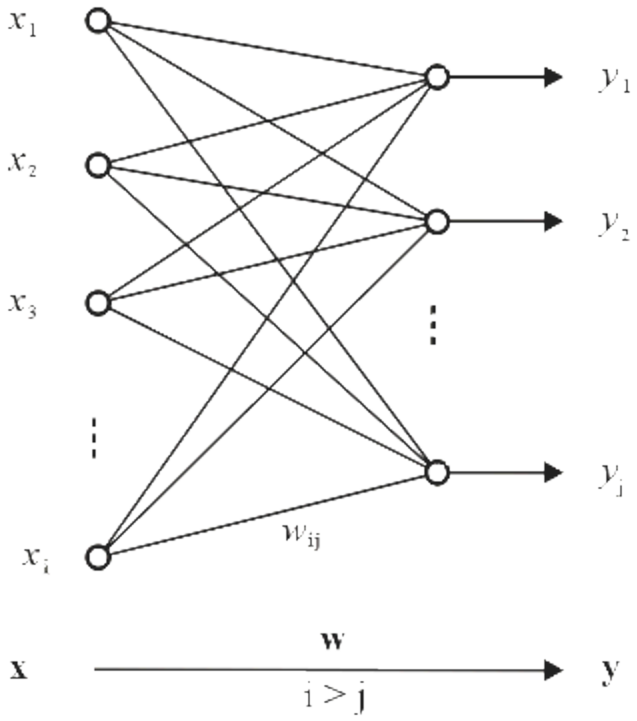
A neural network is an interconnected system of simple processing elements whose behavior is inspired by the activity of biological neurons. The intelligence of the network is derived from the way in which these artificial neurons are linked; which neurons are connected and what the strength of their connection is. These links, referred to as *weights* are adaptively adjusted though out a training period allowing the network to learn the correct output vectors which correspond to certain input vectors.



A typical neural network architecture. Taken from www.gamedev.net/

Hebbian learning is based on the observation that when one neuron contributes to the firing of another neuron the pathway between the two neurons is strengthened, and when it does not contribute to the firing the connection is weakened.[1] This occurs in the human cortex in the form of long-term potentiation (activation) and long-term depression (inhibition) of neurons associated with long-term memory storage.[2] This activity is simulated computationally by increasing or decreasing variables representing synapse strengths based on whether or not the neurons they connect to are both firing. Hebbian learning forms a self-organized internal model of statistically salient aspects of the external environment.[3] This is appealing because it requires no external "teacher" to inform the network as to what is the "right" or "wrong" answer; rather the network learns

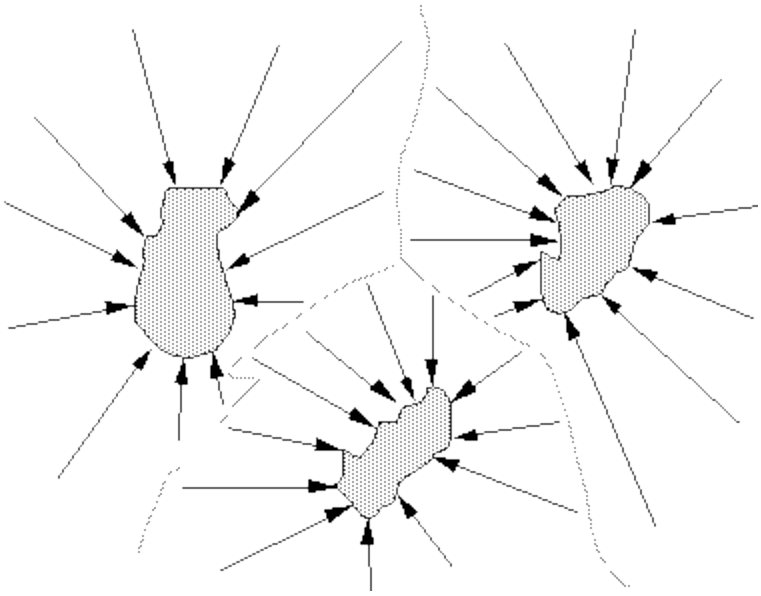
associations through experience.



A Hebbian architecture. Taken from www.ivorix.com

Memory is the persistent effect of experience[4] and attractor networks are the computational correlate of human long term memory.[5] Utilizing techniques inspired by Hebbian learning attractor networks remember by adjusting the weights between neurons based on input to the system. Attractor networks form content-addressable memories, meaning they remember information based on characteristics of the information itself.[6] This stands in contrast to the traditional form of random access computer memory which stores and recalls information by an arbitrary address. Content-addressable memory is desirable because it approximates the human ability to remember based on partial and degraded input. For example, we can easily recognize faces of people we know well from noisy and distorted photographs.

An *attractor* is a state or output vector in a system towards which the system consistently evolves toward given a specific input vector. The *attractor basin* describes the set of input vectors surrounding the learned vector which will converge to the same output vector.[7] This basin is what allows the attractor network to recall learned vectors from degraded input. For example, if the network has learned to associate the input vector 1001 with the output vector 1100, the degraded input 1000 should also converge to the learned output vector 1100.



Dynamic system S

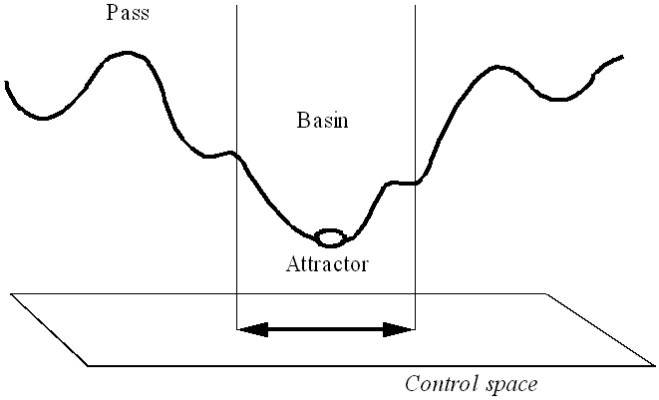
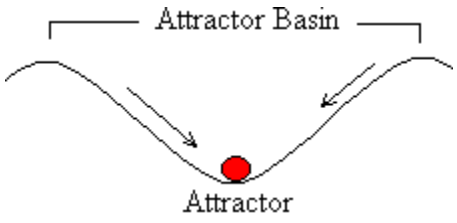


Figure 2

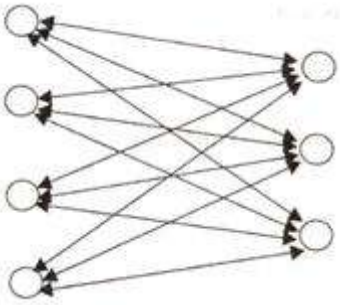


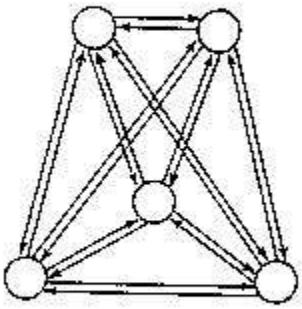
Depictions of attractor basins. From top: http://pespmc1.vub.ac.be, www.revue-texto.net, http://richardbowles.tripod.com,

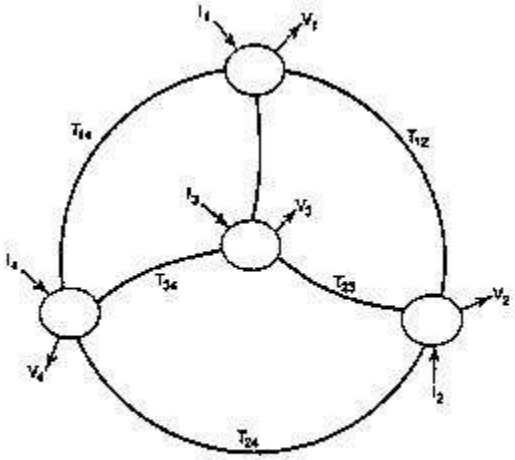
A Bi-directional Associative Memory is a particular type of attractor network which learns associations between two fully interconnected layers of neurons (X and Y). The network is described as a matrix of signed integer weights determined by the set of patterns it knows. For standard applications the weights can be worked out in advance of implementation according to a simple formula. The basic algorithm as described by Luger is:

> 1. Apply an initial vector pair (X,Y) to the processing elements. X is the pattern for which we wish to retrieve an exemplar, Y is randomly initialized.
> 2. Propagate the information from the X layer to the Y layer and update all the values at the Y layer.
> 3. Send the Y information back to the X layer, updating all the X units.
> 4. Continue the preceding two steps until the two vectors stabilize.[8]

For example, if we take a neural network with an array of 8 light sensors and 8 touch sensors and we expose the network to two input vectors, 11110000 and 10101010 repeatedly it will learn to associate these vectors together by increasing the weights of the neurons firing 1 together and decreasing the weights of the neurons firing 0. If the touch input is held constant and the light input vector of 11110000 is sensed the network will recall the touch vector of 10101010. Likewise if the light input is held constant and the touch vector of 10101010 is received the network will remember the light vector 11110000.







Simplified 4-neuron
Hopfield network

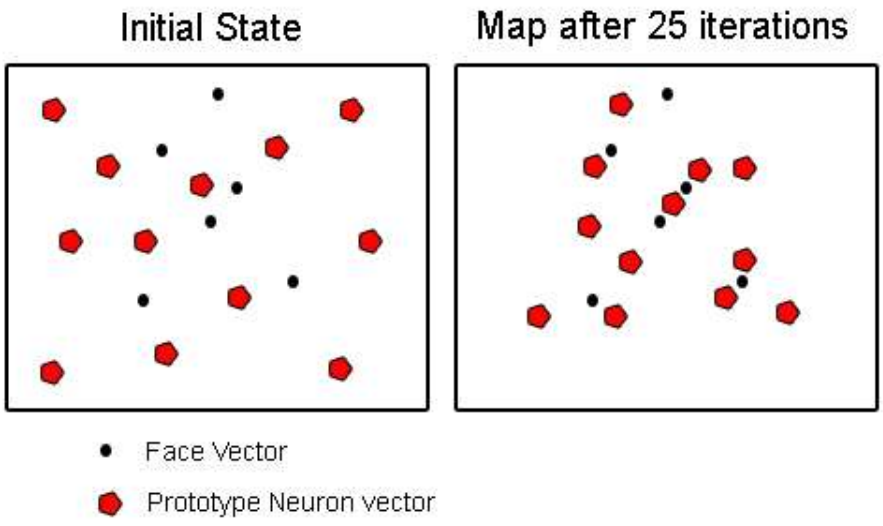
From top: Bidirectional Associative memory. Taken from www.molgorod.cap.ru

2 Hopfield Network depictions. Taken from http://www.benbest.com

Auto-Associative Memories, also referred to as a Hopfield networks, use the same framework as BAM, only in this case the X layer of neurons and the Y layer of neurons are the same. Auto-Associative memories are generally used for content-addressable pattern retrieval as opposed to associational memory.[9] [10]

Self-Organizing Maps (SOM) are based on the winner-take-all algorithm developed by Tueuvo Kohonen in 1984 and extended by many others since.[11] Referred to as competitive learning, in most SOM models only one output neuron is active at a time and neurons "compete" for activation.

Like the Hebbian learning described above SOMs are a form of unsupervised learning, they require no external teacher and have no ideal notions of right and wrong answers. They are a form of feature extraction and clustering, self-organizing over repeated exposure to input data to form groups and category descriptions of the information.



A SOM is generally a two layer feedforward network of neurons, consisting of completely interconnected inputs and outputs. It works with continuous valued input vectors, in contrast to most neural network architectures which are designed for discrete binary values. Each output neuron is characterized by the vector of weight values connecting it to the input layer.

The network is initialized with random weights which form a set of "prototype vectors".[12] When an input vector is applied to the system the output neuron with the highest value is declared the "winner" and its weights are updated to more closely match the input vector. The winning neuron is considered to be part of a "neighborhood" of neurons with similar weight vectors. After the winner is found every neuron's weights are updated proportionally to how close they are to the winner. This is very similar to the Hebbian method of updating weights by reinforcing existing similarities between input and output. Over time and repeated exposure to input the weights of the output neurons come to reflect the structure of the input data. Some neurons match input values exactly, others assume intermediate positions between vectors. This allows the network to extract *features* from the input data, to group data points into specific categories and to perceive new or noisy input data through the lens of what it knows. [13]

3.2 Principle Component Analysis

Principal Component Analysis (PCA) is a commonly used statistical method of correlated data analysis. Given large, high dimensional, and intuitively intractable data sets, PCA can effectively and quickly reduce the dimensionality of the data thereby compressing it and extracting its specific characterizing features, or *principle*

components. [14] [15] [16] PCA has been explored in neural networks and machine vision contexts and a variation called eigenface (or eigenimage) analysis has proved useful for tasks such as face and handwriting recognition.[17]

The Hebbian learning method described above extracts the first principal component of the input data set.[18] Variations of Hebbian learning including Sanger's rule and Oja's rule are capable of extracting every principle component of the dataset in order of relevance.[19]

Principal component analysis of two-dimensional data. Line shows the direction of the first principal component. Taken from www.cis.hut.fi

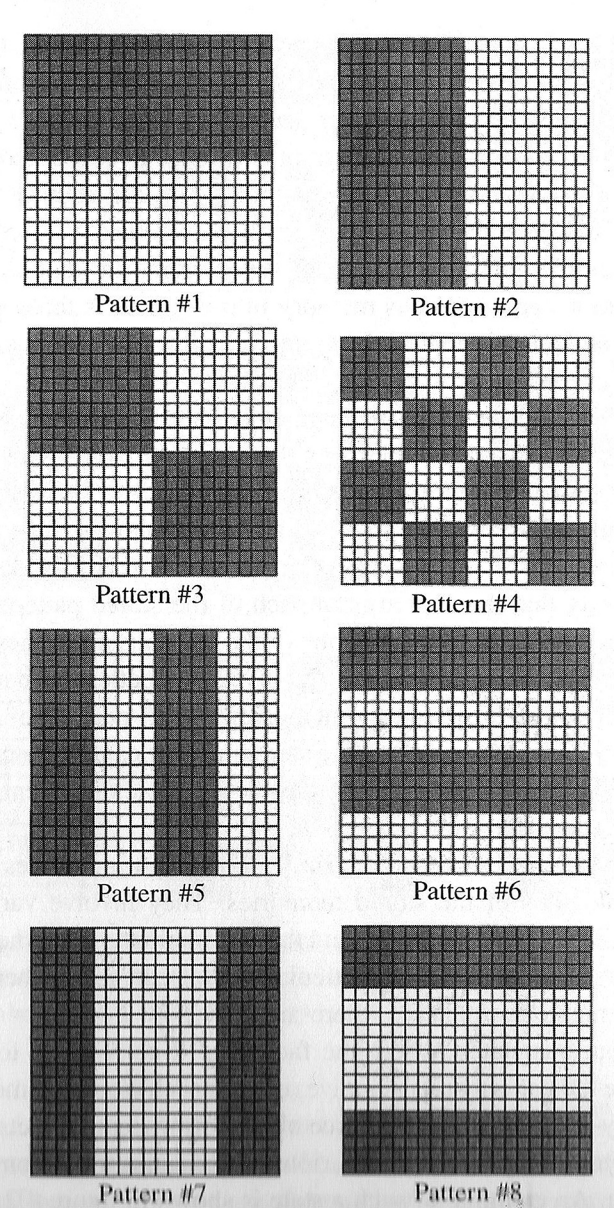
3.3 Psychology

The initial inspiration for my thesis project comes from George Christos' book *Memory and Dreams*. Christos describes creativity from a computational neuroscience perspective, as "spurious memories"; false attractors in the overlapping distributed architecture of memory. In his words:

> "Memories change with time because of their interaction with each other -that is, through spurious memories. Memories are combined together or interact with each other because they share the same synaptic connections. Spurious memories are the consequence of this sharing... What makes the brain particularly adaptive, associative, and capable of generating new creative states is that it can generate its own memory states, which were not intentionally stored in the system."[20]

He supports a slightly modified version of the Crick and Mitchinson hypothesis that "we dream to forget"; that dreaming is necessary to weed out spurious memories accumulated during the day, allowing relevant associations to grow stronger. Christos describes his experimentation with Hopfield Neural Networks, and his success in achieving a very simple but exciting demonstration of visual creativity derived from spurious memories.

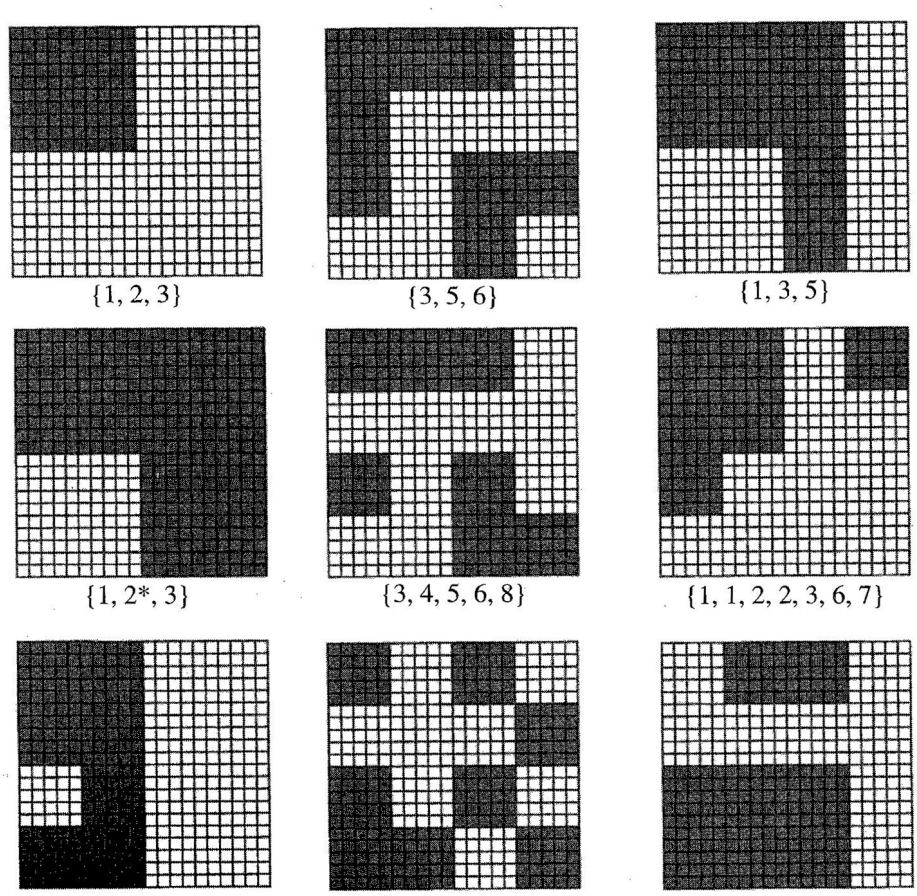
Christos' hypothesis about creativity stems directly from his work with Hopfield neural networks. Attractor memories in general are capable of storing about 15% of their neurons in patterns. In other words a neural network with 100 neurons can store 15 patterns reliably.[21] One of the interesting inherent characteristics of these networks is that in addition to storing the desired vectors other, spurious attractors also are "learned". This means that the network may remember a pattern or association that it did not actually experience.



Learned Pattern Vectors[22]

Christos' experiment uses a Hopfield network with four hundred neurons to store the eight simple orthogonal patterns you can see in the figure above.

He then applies random unlearned input vectors to the network to explore what other, hidden attractors exist. Some of the patterns he discovered are shown in the figure below.



Spurious Pattern Vectors[23]

The spurious memories are visibly synthesized from components of the learned vectors. They differ from the learned patterns but feel contextually relevant, obviously not random. Intuitively they look like what I describe as "everyday creativity", the generation of something new that is not historically exceptional; the kind of creativity we use every time we speak to generate new patterns of words derived from our experience with speaking.

John Antrobus disagrees with memory consolidation theories which attempt to explain the purpose of dreaming, stating that there is really no evidence one way or the other. His take on dreams is that they are nothing more than random firing in the

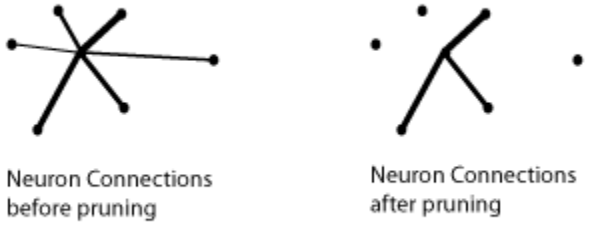
brainstem and the brain's attempt to interpret this information, stressing the importance of context on perception. In his article "Thinking Away and Ahead" in the Santa Fe Institute publication "The Mind, The Brain and Complex Adaptive Systems" Antrobus sites an experiment from Bill Dement's thesis research, wherein a subject in Rapid Eye Movement (REM) sleep is lightly sprayed with water. When awoken they describe dreams which incorporate this environmental stimulus in some way, for example they are standing in a room talking to a woman and suddenly the roof starts leaking, splashing them with water. This provides a perfect example of *context dependent creativity*; the mind's necessity to create an explanation within its frame of reference, derived both from previous sequential thoughts and sensory information.

Antrobus demonstrated this context dependent incorporation of stimuli using a recurrent backpropagation neural network. He began by teaching a standard network 28 binary image sequences composed of 4 subsequences such that the first two bits in the sequence specified the subsequence or context. Simulation of dreaming was accomplished by feeding the output of the network back in to its input, akin to what happens during REM sleep. Antrobus found that just like the subject who incorporated a spray of water into their dream, if he primed the neural network in dream mode with a novel, unlearned image the network would respond with a context-appropriate interpretation of that image.

Ralph E. Hoffman was one of the first psychologists to successfully model delusions and schizophrenia using a neural network.[2425] What intrigued me about Hoffman's model is that it is practically identical to Christos' model of creativity. They both studied spurious attractors in Hopfield neural networks using simple geometric patterns, the only difference is that Hoffman's hypothesis was based on excessive pruning

of connections between neurons in the network. His theory was that schizophrenia arose from having too few cortical connections between neurons. This idea was based on evidence from comparative postmortem studies where it was discovered that schizophrenics have fewer connections between neurons in their pre-frontal cortex than non-schizophrenics.[26]

Image of pruning. Taken from



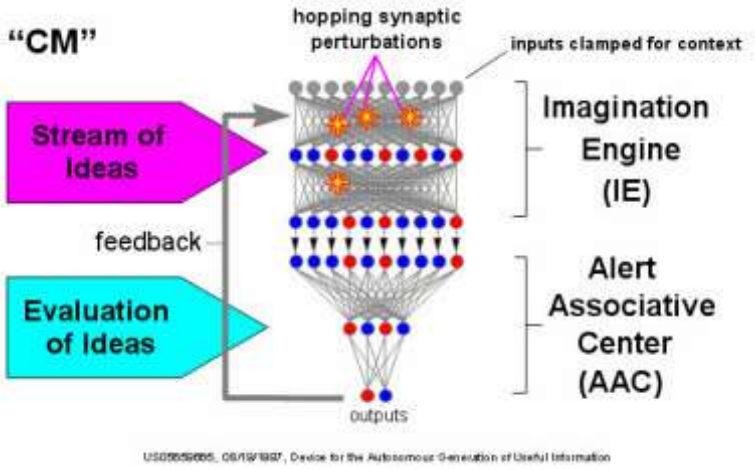
Hoffman simulated this loss of conne
teaching it to perform pattern recognition, an
network. He found that the model would digr
outputting patterns that were not only different from the input it was experiencing, but were also different from any input it had ever experienced. This provided a plausible explanation of how delusions might arise from the structure of the schizophrenic brain.[27][28]

3.4 Imagination Engines

Stephen Thaler is an entrepreneur with a patented recipe for what he calls a "Creativity Machine". Also working with neural networks, his approach is that of the computer scientist rather than a neuroscientist. Instead of trying to exactly simulate the brain Thaler's patented network structure abstracts biological ideas in a computationally pragmatic way. Starting with a multi-layer backpropagation neural network which is trained in a certain problem domain, his approach freezes the inputs to the network and proceeds to randomly perturb connections between neurons in the hidden layers. He then

feeds the output of this network to the input of another neural network which is taught by experts in the problem domain to recognize novel and salient outputs.[29] Thaler's technique appears to be remarkably successful, and he credits his "Creativity Machines" with unique and pertinent inventions in fields ranging from dental hygiene to music.

Image of Thaler's network architecture. T **"CM"**



Though the ingenuity of his project st
creativity, the element I find interesting is the
random firings in the brainstem which occur
holding sensory input still and randomly appl
looks a lot like what both Christos and Antro

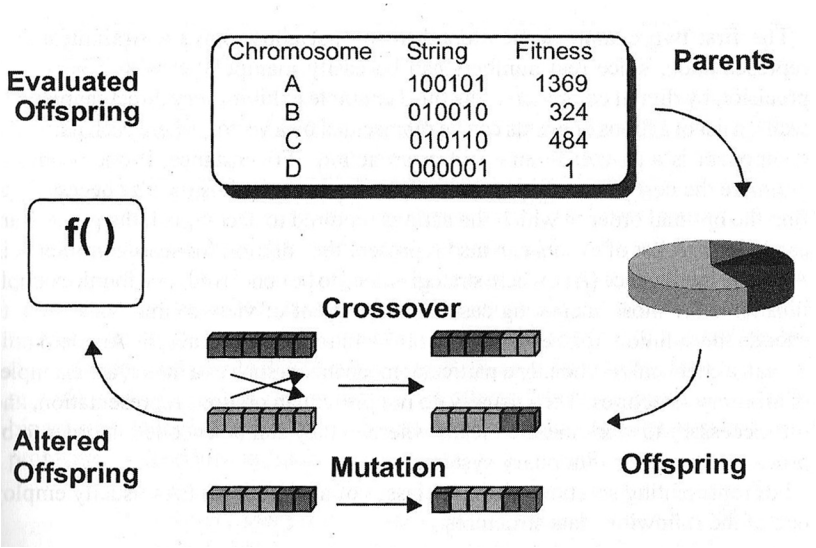
3.5 Evolutionary Computation

Evolutionary computation, like neural network theory, is a sub-section of machine learning. It provides an emergent model of learning based on the biological process of natural selection. Beginning with a specific problem and goal, an evolutionary algorithm will evolve a population of candidate solutions which compete on the basis of their fitness, or proximity to the goal. The algorithm is iterated such that each time through the best candidate solutions breed to form the next generation. Utilizing operations inspired by biological cross-over and mutation, evolutionary algorithms are capable of effectively finding solutions to problems with search spaces which are too large to tackle using an exhaustive search.[30] Like biological evolution, they do not guarantee to result in the *best* solution to the problem, rather they promise a solution which is *good enough*, which meets the minimum criteria for success.

The genetic algorithm was formalized by John Holland in his 1975 book *Adaptation in Natural and Artificial Systems*. The algorithm is simple:

1. Create an initial population of candidate solutions
2. Evaluate the initial population
3. While no member of the population meets the criteria for success:
    - -Select individuals into a mating pool
    - -Create a new population using crossover and mutation
    - -Evaluate the new population[31]

The difficulty with genetic algorithms is generally finding a good mapping between the genotype and phenotype of the population. Genetic algorithms historically use a binary representation of the possible solution as DNA. The DNA representation must be able to adequately describe a solution and be modular enough to progress using crossover and mutation.



Basic Flow of an Evolutionary Algorithm[32]

There are five dominant methods of selection used in genetic algorithms to determine which members of the population will be allowed to breed: proportional, linear ranking, exponential ranking, tournament, and truncation. Truncation selection is the simplest method. If candidate solutions meet a minimum threshold of fitness they are placed into the mating pool, otherwise they die out. Proportional selection allows candidates a probability of mating which is directly proportional to their fitness by copying them into the mating pool that number of times. Tournament selection acquires a mating pool by repeatedly choosing two individuals randomly and picking the fittest one to breed until the mating pool is full. Linear rank selection creates a linear distribution

between the fittest and the least fit candidate and then proportionally adding them to the mating pool. Exponential ranking does the same thing but with an exponential fitness distribution. [33]

Once individuals have been selected into the mating pool, breeding begins by selecting two candidates at random to become the parents. Crossover consists of randomly choosing one or more splicing points in both parents DNA and swapping alternate pieces. The process results in two offspring which are a combination of both parents. The children are then subject to a probability of mutation and finally evaluated for their own fitness.

There are three popular variations of the genetic algorithm, evolutionary programming, evolutionary strategies, and genetic programming. Evolutionary programming was first used in 1962 by Lawrence J. Fogel. Unlike the genetic algorithm, it does not specify how candidate solutions should be represented (their DNA) and it does not utilize crossover. It usually uses the truncation selection method. In 1965 Ingo Rechenberg and Hans Peter Schwefel came up with Evolutionary Strategies. They invented the idea of multi-point crossover, and used tournament selection.[34] This technique was first used to progressively improve a single candidate solution. The fourth variant was described by John Koza in his 1992 book *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Genetic programming is used primarily for evolving computer programs. It is generally coded in the LISP programming language and represented using tree diagrams.[35]

3.6 Evolvable Hardware

Evolvable Hardware(EHW) is a new discipline emerging from the development and proliferation of Field Programmable Gate Arrays (FPGAs), and to a lesser extent Field Programmable Analog Arrays (FPAAs). It is based on the success of evolutionary programming in software development and attempts to port the genetic algorithm to hardware.

FPGAs are user defined circuits in a chip. Consisting of a grid of Configurable Logic Blocks, Input/Output Blocks, Multiplier blocks, Block RAM, a Digital Clock Manager, and programmable interconnects, an FPGA can be programmed to create most basic digital circuits, providing a cost-effective alternative to Application Specific Integrated Circuits (ASICs).[36] Their parallel structure and abundance of input/output ports make them an ideal medium for evolutionary design.

There are two different methodologies for evolving hardware, intrinsic evolution which takes place completely in hardware, and extrinsic evolution which runs as a software simulation. Circuits which are evolved extrinsically are only implemented in hardware after the goal circuit is reached. Intrinsic evolution works by evolving the configuration bitstream for the device directly in hardware. This allows for a greater variety of possible circuits and also enables the evolution of circuits which take advantage of their environment and the silicon specific properties of the device they evolve in. Conversely, circuits evolved extrinsically are often more robust and portable and are less prone to generating "illegal" circuit designs which damage the chip itself.[37] [38] [39]

Cartesian Genetic Programming (CGP) developed by Julian Miller and Peter Thomson is an example of a representation scheme for evolving hardware. The device to be configured is represented as a matrix of nodes arranged in columns and rows, and is

encoded into DNA as a string of integers. Each node is represented by a gene in the DNA sequence consisting of 3 integers. The integers stand for what type of gate the node is and what inputs it is connected to. The gates are standard logic gates, AND, OR, NAND, NOR, XOR, XNOR, and NOT. The inputs are the physical inputs to the chip along with the outputs from every node in the matrix. Columns specify the sequential order of signal flow in the system and nodes are allowed to connect to the outputs of other nodes in the columns preceding them. A levels back parameter specifies how many columns forward a node is allowed to connect. By representing the circuit as a matrix of nodes CGP preserves Mendelian heredity, allowing parent circuits to pass specific genes coding for particular input/gate combinations to their offspring. [40 41 42]

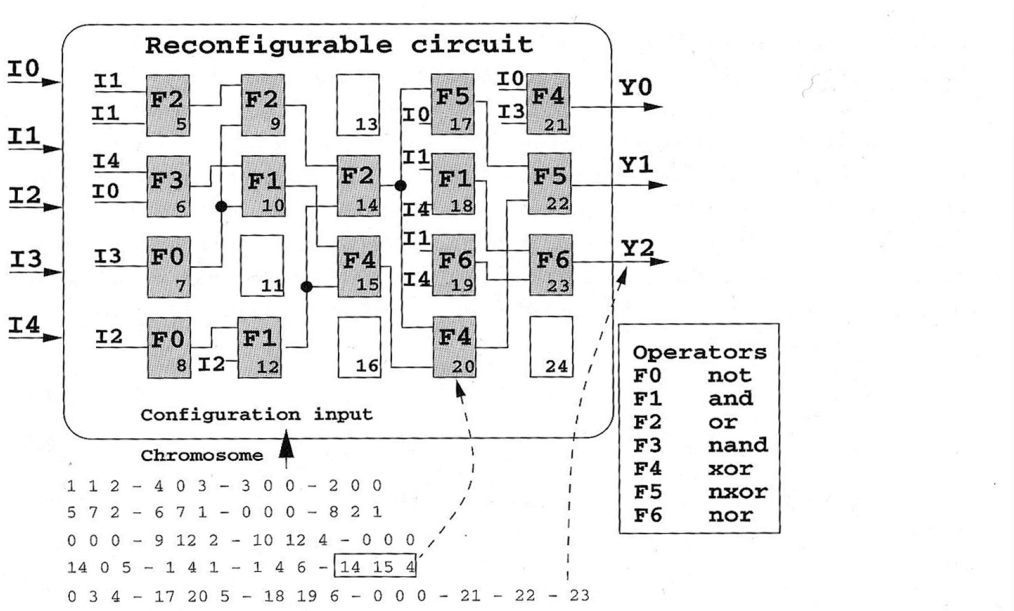
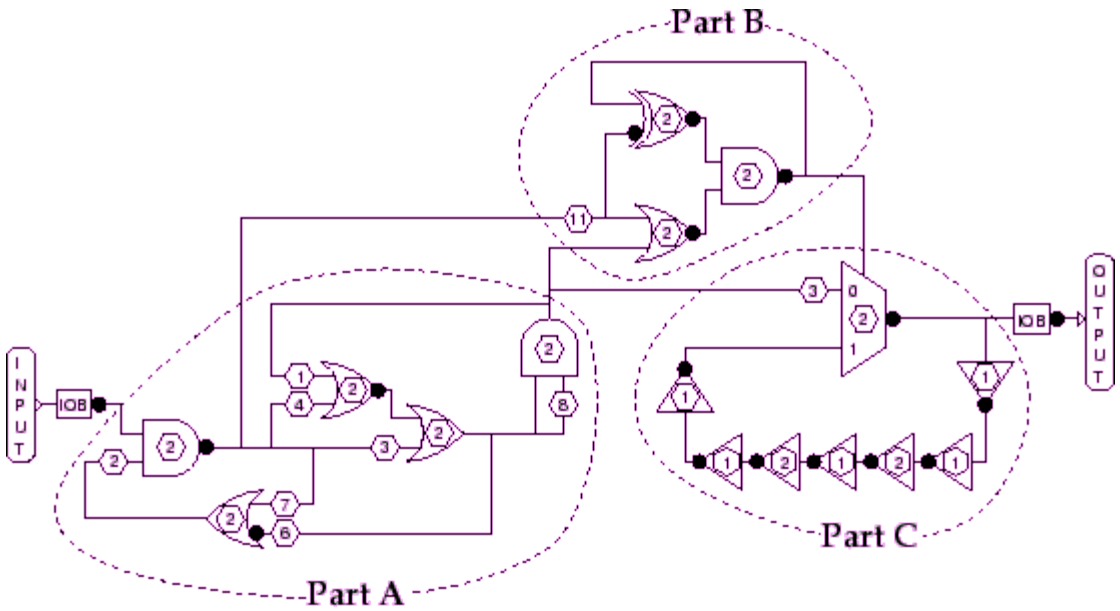


Fig. 4.1. An example of the reconfigurable circuit and its configuration according to CGP: $n_r = 4, n_c = 5, n_i = 5, n_o = 3, n_n = 2, n_f = 7, L = 1, F = \{\text{NOT, AND, OR, NAND, XOR, NXOR, NOR}\}$. The circuit inputs are indexed 0–4 and the programmable nodes are indexed 5–24. Three genes define the configuration of every node. Only utilized nodes are marked. The last three genes of the chromosome determine the connection of the circuit outputs

Cartesian Genetic Programming. Taken from Sekanina, p. 43

Historically the field of evolutionary hardware began with Louis and Rawlins paper *Designer Genetic Algorithms: Genetic Algorithms in Structure Design* describing a technique for evolving circuits like building blocks.[43] The first major publicized success in the field occurred in 1996 when Adrian Thompson successfully evolved a circuit which could discriminate between a 1 kHz and a 10 kHz tone.[44] This experiment was particularly exciting because the evolved circuit was more efficient than traditional tone discriminators in terms of resource use, and its design completely baffled engineers.



Evolved intrinsically on an FPGA, the circuit made use of unconnected components, and could not be ported to a different chip or even a different temperature! It was completely site-specific.

Representation of Thompson's evolved tone discriminator circuit.[45]

This ability to evolve unique circuits which can adaptively reconfigure themselves

over time and are integrally related to their environment and their own physical structure (body) makes evolvable hardware an interesting new direction to pursue for research into embodied electronic creativity.

4.0 The word "Creativity"

> *"The problem of creativity is beset with mysticism, confused definitions, value judgments, psychoanalytic admonitions, and the crushing weight of philosophical speculation dating from ancient times."*
> -Albert Rothenberg

The word "creativity" did not exist for most of human history but its origin lies in the word "creation". It begins with the cosmology of the ancient Mesopotamians and undergoes perpetual change, transforming in relation to dominant cultural ideas of the self. This process of transformation and cultural re-definition has left the word as a confused palimpsest of meanings and interpretations.

4.1 History

In the Hebrew bible there is a clear linguistic distinction between God's creation and human creation. The verb "bara" generally translated as "created" means literally "to carve out" and later "to perfect". When God's creation is referred to it is in the *Qal* or simple form of the verb, but when man's work is being described it is using the intensive form of the verb. This connotes a sense that God's work is immediately perfect, it takes

no effort, whereas man must labor.[46] [47]

The word "creation" was not applied to the work of humans until after the Renaissance. In 1603 Shakespeare was the first to linguistically transfer the meaning of the word "creation" from God to man:

"Or art thou but A dagger of the mind, a false creation,

Proceeding from the heat-oppressed brain?"[48]

This was the first use of the word "creation" referring to an original production of human intelligence or power.[49] Additionally, Shakespeare draws a direct connection between the mind and the brain. This quote implies a holistic view; a direct relationship between imagination or hallucination and the physical brain. Nonetheless, it was not until 1875 that the word "creativity" was first used in print.

It is fitting that the word "creativity" was coined to describe the man who first used the word "creation" to apply to a work of the human mind. In his *History of Dramatic English Literature* Adolphus William Ward describes "The poetic flow of (Shakespeare's) spontaneous *creativity*."[50] Fifty years later the word emerged in French and Italian, and finally after World War II in common English.[51]

Ferdinand Helmholtz was the first person to describe the "creative process" as a concrete series of steps: Saturation, Incubation and Inspiration. Graham Wallas later published Helmholtz's account and added a fourth step of verification. In 1950 J.P. Guilford, president of American Psychological Association called upon psychologists to study creativity, and they heeded his request.[52] From this point on creativity became a subject of investigation in psychology as well as neuroscience, computer science, art philosophy and business. Of these studies, theories, and marketing techniques there came to exist countless divergent definitions and concepts.

4.2 Definitions

Today, we discuss intelligence and creativity as goals for computer science. Through an increasing understanding of the physical nature of the brain we strive to comprehend and to simulate the mysteries of the mind. I believe that these goals are possible, but the *creation of creativity* requires a distinct and specific definition of meaning.

The concept of creativity has a long history of usage in diverse contexts, and it is exactly this pastiche of connotations that makes its definition and explanation so controversial. It is what Ludwig Wittgenstein would call a confusion of language games; a conflation of the various ways in which we *use* the word.

> *"A 'picture' held us captive. And we could not get outside*
> *it, for it lay in our language and language seemed to repeat*
> *it to us inexorably"*[53]

I propose a Wittgensteinian clarification of the *idea* of defining creativity. In my own experience I notice confusion arising when I begin to speak of ideas like "machine creativity". I am at once accosted with arguments describing the fundamentally *human* nature of the very *idea* of creativity, its subjectivity and its cultural relativity.

Given the historical legacy of the word I can not disagree with any of these arguments. I embrace their contextual relevance, but I must explain that I am in fact describing something different.

When I speak of "machine creativity" people have at least a fuzzy notion of what I mean. The phrase has sensical use as a term; it is in fact a meaningful concept. If creativity is by definition a human phenomenon then how can we speak of creativity in

relation to machines, or for that matter, animals? It would be nonsense. The answer, I will argue, is that when we speak of machine creativity we are using the rules of a different language game.

> *"Consider for example the proceedings that we call 'games'. I mean board-games, card-games, ball-games, Olympic games, and so on. What is common to them all? ... if you look at them you will not see something that is common to all, but similarities, relationships, and a whole series of them at that... I can think of no better expression to characterize these similarities than 'family resemblances'."*[54]

There are at least two easily discernible language games in which we use the word "creativity", they are what Margaret Boden calls historical (H-creativity) and psychological (P-creativity).[55] There are "family resemblances" between the way the word creativity is used in both contexts, but as games they consist of fundamentally different rules of use. The historical perspective speaks of the culturally unique phenomenon of creativity. Individuals transform cultural information and if their output is *deemed valuable* in its field then it is accepted into the respected canon.[56] This view stresses the importance of environment on the creative individual. It views creativity as a cultural loop between the individual, their peers and the domain within which they are working; with any of the essential pieces missing the creative act is neutralized.

The psychological perspective of creativity is more concerned with examining its traditionally *ex nihilo* reputation. Whether or not an individual's creative contribution is historically new or not, the question is how they were able to come up with an idea which

was *new to them*.

> *"Whether or not computers can really be creative they can do apparently creative things and ... considering how they do so can help us understand how creativity happens in people"*[57]

This perspective on creativity assumes a materialist standpoint, taking as a basis that something simply cannot arise from nothing; it is physically impossible. Rather, the human capacity for generating new ideas appears to be an emergent property of the complex system of the brain arising from the way in which the brain stores memories in an overlapping, distributed, and associative fashion.[58]

Historical creativity describes the extraordinary accomplishments of people like inventors and artists. Psychological creativity is *everyday creativity*, the kind of creativity all humans use in constructing new sentences, walking over new terrain, or preparing food.

It is difficult to discuss the concept of creativity because it has so many different scopes of use as a word. My own definition is that creativity occurs when an output is generated which was not explicitly learned. This stems from the psychological perspective of creativity and draws heavily on neural models of computation. By distinguishing the particular language game in which I am using the word I believe it becomes possible to define and therefore to investigate.

5.0 Methodology

Beginning with theoretical research my thesis project progressed to software modeling, electronic hardware evolution, and finally ended up back in software as a creative face generating neural network system. Along the way I became intimately acquainted with neural networks, philosophy of Artificial Intelligence, Field Programmable Gate Arrays, Evolvable Hardware, Principle Component Analysis, as well as the psychology and neuroscience of memory and perception.

When I first started thinking and talking with others about my idea to create a creative machine, I was immediately confronted by the philosophical difficulties of speaking about a word as subjective and specific to human culture as "creativity" in relation to a machine, or in any kind of quantitative, empirical fashion. I soon realized I would have to figure out what I meant by the word and come up with a solid definition of my own. I arrived at "The generation of an output which is not explicitly learned" from my initial research into neural networks, the biology of memory, and Artificial Intelligence. It is a definition which is open enough to include machines but exclusive enough to exclude most machines in existence today. It is a memory oriented definition derived from the idea that creativity is an emergent property of the complex overlapping system of long term memory in the brain.

From my research I have found two ingredients that I believe are essential to creativity in any medium:

1. Attractor based memory
2. Ambiguous input

Attractors were discussed in section 3.1 and refer to what we commonly would

just describe as memories. From an initial state in a dynamic system, attractors are the final state toward which the system evolves given a specified range of input.[59] The idea of attractors stems from thermodynamics but has been applied to study of the brain and neural networks as well. Attractor dynamics are essential to neuronal models of memory,[60] and Christos' theory of spurious memory (section 3.3) is a natural correlate of this model.

Attractors and in particular spurious attractors provide a way to start thinking about the biological foundation of creativity, and the possibility of creativity in other mediums, but they aren't enough on their own. In order for attractors to come into use an ambiguous input must be present. It is the brains attempt to reconcile unexpected input data that would lead it to one of these spurious attractors. This is where Christos' theory overlaps with Thaler's model of a noise based creativity machine, and Antrobus' model of a dreaming neural network. Thaler introduces noise to get his networks to think slightly differently from what they have been taught, and Antrobus discusses the brain's attempt to make sense of random neuronal firings during sleep. When we combine an attractor memory with random input what we get is exactly the networks attempt to make sense of that input, and whatever attractor the input is closest to is what it will be recognized as or associated with. Whether it is random noise neurons firing during sleep or looking at an ambiguous image like an ink blot or a cloud, the human brain is always attempting to derive meaning and form from what it perceives and I believe that this ability is central to creativity.

With these ideas in mind, I proceeded to build the neural network models described by Christos, Antrobus and Thaler to see what results I came up with and how good it looked after delving into the math and the code. I was glad to find all their results

completely repeatable and soon was looking for a way of merging these ideas into one neural network architecture which would be capable of handling information more complex than the simple examples described by Christos and Antrobus.

Meanwhile, I was becoming fascinated by the field of evolvable hardware (section 3.6) and gradually determined to incorporate this into my thesis as well. What better way to create a population of unique electronic brains than by actually evolving the hardware components? I purchased the necessary FPGA board to begin my experiments, taught myself the hardware description language necessary to program it, and developed a genetic algorithm system to evolve hardware configurations extrinsically.

Despite my excitement I eventually realized that evolvable hardware was simply not going to work. The hardware and documentation were designed to discourage the kind of low level hacking required to do intrinsic hardware evolution, and my extrinsic evolution was moving along very slowly. After many unanswered emails to academics in the field and the FPGA manufacturer I regretfully decided I would be able to accomplish more, quicker in software.

After this foray into hardware I resumed my neural network research. I became interested in working with facial images after reading about the brain's incomparable ability to recognize faces and tendency to see them in everything.[61] My piece then could be viewed as an artistic interpretation of the specialized brain hardware devoted to face perception. It carries a dual meaning also, by subverting algorithms which might ordinarily be used for surveillance and control and reworking them as instruments of what John Cage called, "purposeful purposelessness".

I began by experimenting with Bi-directional Associative Memories and then Hopfield networks to see what kind of inputs they could work with. I wanted to work

with high resolution grayscale images of faces, but soon found that neither network was designed to accommodate my needs; they required discrete binary input data, not continuous grayscale values, and the Hopfield network produced a combinatorial explosion of the number of connections needed when I increased the pixel resolution from Christos' 20x20 pixel grids. Additionally I discovered the Hopfield network requires *uncorrelated* data in order to properly function, and faces are a *highly correlated* dataset; our faces are vastly more similar in appearance than they are dissimilar. I knew I had to find a different model, one that was designed to handle continuous, correlated data without requiring an exponential increase in the memory needed for each additional input pixel, but one that would still demonstrate the phenomenon of spurious attractors.

Around this time I began reading about Principal Component Analysis and eigenface analysis in relation to Hebbian learning. I learned that I could reduce the dimensionality of my input facial images extracting their principle components and found a handful of mathematical models that could accomplish this. I discovered an open source eigenface code library at Rice University which proved invaluable in helping me understand the math behind the system. I experimented with both neural and standard models of analysis and eventually developed on a hybrid model of my own which built on the open source model from Rice.

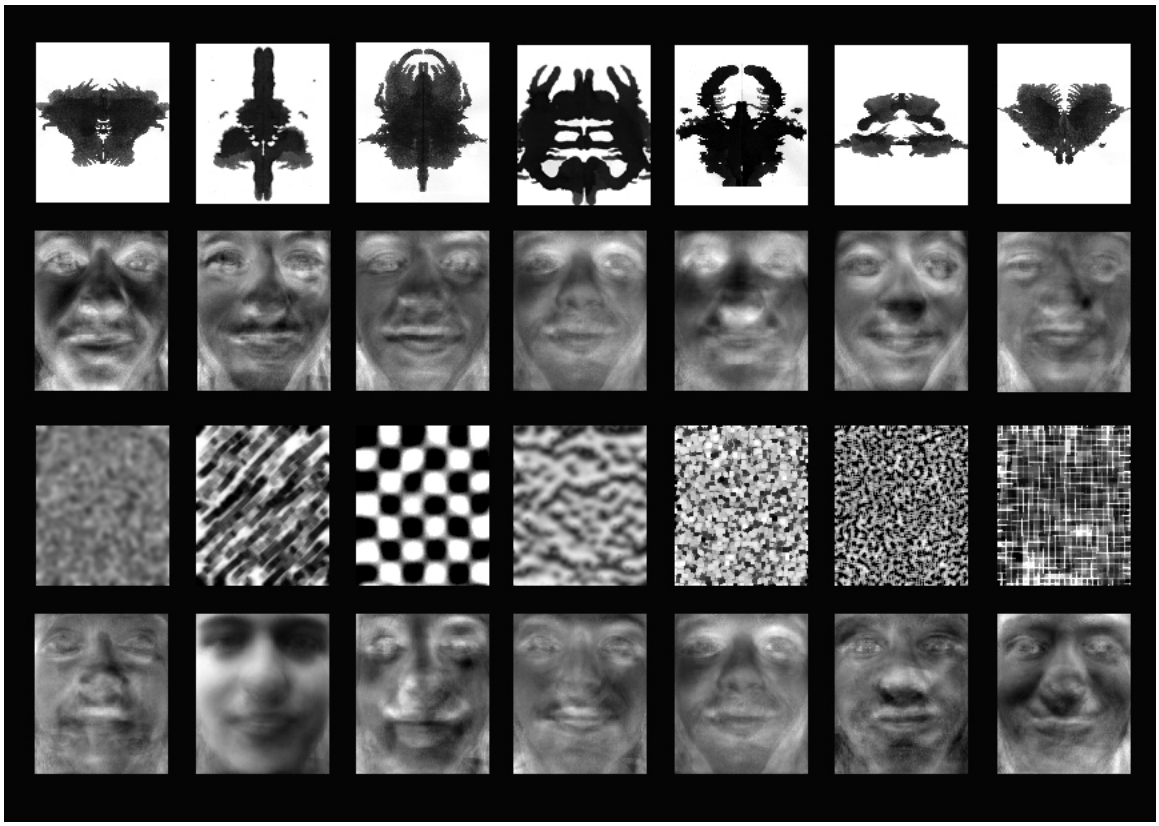
Principal Component Analysis solved my dimensionality problems but did not help me with the problem of continuous values. I went back through my neural network resources and found a different model, the self-organizing map (section 3.1) which was specifically designed for exactly the kind of data I wanted to work with. After some preliminary software experiments I realized that with the right parameters and architecture the neighborhood updating technique used in the self-organizing map created

spurious attractors very similar to those found in the Hopfield model.

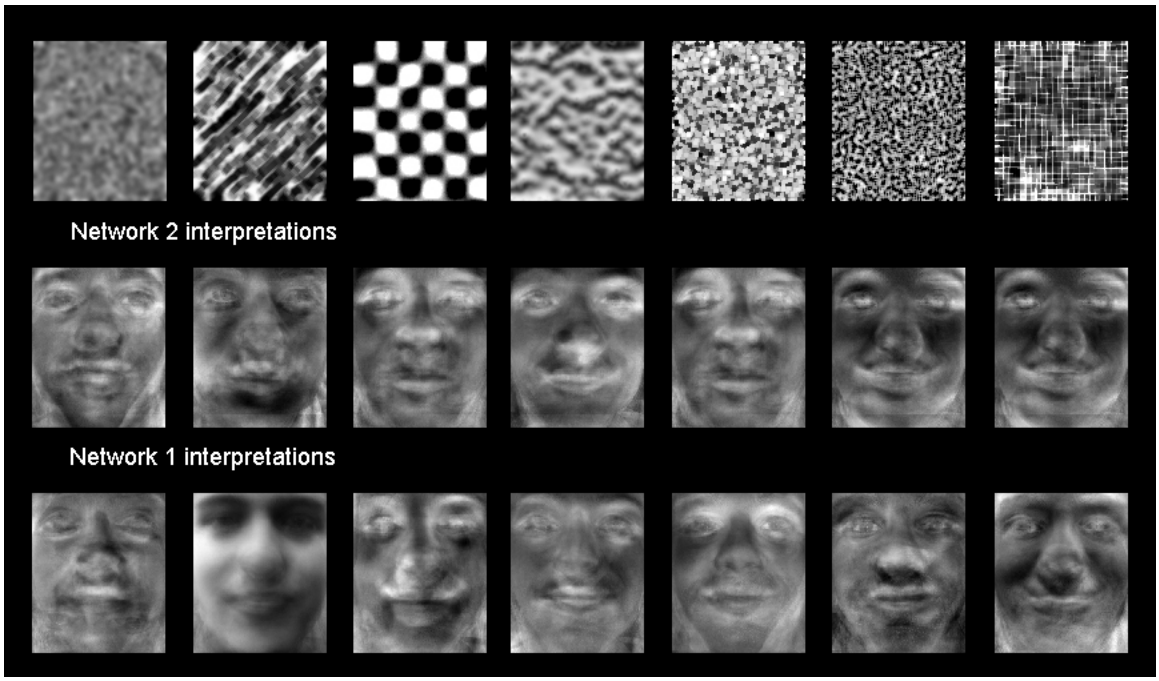
My next step was synthesizing the two into one cohesive network structure which utilized Principal Component Analysis to form an inductive bias for face perception and a self-organizing map to categorize the data. I then added a bidirectional capability to the network which allowed it to generate images as well as categorizing them, and finally added a feedback capability which allowed the network to go into a recurrent mode inspired by Antrobus' model of dreaming.

A selection of the resulting images can be seen on the next pages.

Sample input faces. Taken from Eigenface project at Rice.



Ambiguous input and the network's interpretation.



A side by side comparison of 2 networks interpretations of the same inputs.

6.0 Conclusions

Beginning with the idea that machines can have a creative life of their own I have developed a system which begins to make that goal a reality. There is much work left to be done, but I believe my project shows that creative machines are possible, both theoretically and physically. It is my hope that this possibility will be explored by others as well and will be used for purposes beyond profit.

With further development I believe creative machines can offer human beings a perspective on human affairs akin to that of an alien culture by providing an outsider interpretation of the people, objects and relations that comprise our society.

If machines are to become creative, not profit-driven innovation engines as in Thaler's work, they need to create *essentially* and purposelessly because their internal structure embodies it and their environment *demands* it. This will initiate a fundamentally new relationship between humans and their technology and will demand a re-examination of human ontology.

7.0 Endnotes

[1] Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (Reading Mass., Menlo Park CA, New York, Don Mills Ontario, Wokingham England, Amsterdam, Bonn, Sydney, Singapore, Tokyo, Madrid, San Juan, Paris, Seoul, Milan, Mexico City, Taipei: Addison-Wesley Publishing Company, 2002), p. 446

[2] Randall C. O'Reilly and Yuko Munakata, *Computational Explorations in Cognitive Neuroscience*, (Cambridge MA, London England: MIT Press, 2000), p.116

[3] *ibid*.

[4] *ibid*., p.275

[5] Luger, p. 457

[6] *ibid.*

[7] *ibid.*

[8] *ibid*., p. 459

[9] *ibid*., p.453, pp.462-463

[10] Hertz, Krogh and Palmer, *Introduction to the Neural theory of Computation* (Reading Mass., Menlo Park CA, New York, Don Mills Ontario, Wokingham England, Amsterdam, Bonn, Sydney, Singapore, Tokyo, Madrid, San Juan, Paris, Seoul, Milan, Mexico City, Taipei: Addison-Wesley Publishing Company, 1991), p. 218

[11] *ibid*, p. 438

[12] Luger, pp. 440-442

[13] Hertz, Krogh and Palmer, pp. 232 - 244

[14] K.I. Diamantras and S.Y. Kung, *Principal Component Neural Networks* (New York, Chichester, Brisbane, Toronto, Singapore: John Wiley and Sons Inc., 1996), p. xi

[15] Colin Fyfe, *Hebbian Learning and Negative Feedback Networks* (USA: Springer, 2005), p. 16

[16] Hertz, Krogh and Palmer, p. 204

[17] http://en.wikipedia.org/wiki/Eigenface

[18] O'Reilly and Munakata, p. 122

[19] Fyfe, pp. 21-25

[20] George Christos, *Memory and Dreams*, (USA: Rutgers University Press, 2003), pp. 101-102

[21] Hertz, Krogh and Palmer, pp. 17-20

[22] Scanned from Christos, p. 79

[23] Scanned from Christos, p. 81

[24] Stein and Ludik, *Neural Networks and Psychopathology,* (Cambridge: Cambridge University Press, 1998), p. 210

[25] http://www.cnbc.cmu.edu/Resources/disordermodels/schizophrenia.html

[26] Ralph E. Hoffman and Thomas H. McGlashan, *Neural Network Models of Schizophrenia*, (Neuroscientist, October 2001), pp. 441-453

[27] *ibid*.

[28] Stein and Ludik, pp. 210 - 212

[29] Stephen Thaler, *A Quantitative Model of Seminal Cognition: The Creativity Machine Paradigm (US Patent 5,659,666),* (Dublin, Ireland: adapted from the Mind II Conference, 1997), pp. 1-4

[30] Luger, pp.469-471

[31] Ricardo Salem Zebulum, Marco Aurelio C. Pacheco, Marley Maria B.R. Vellasco, *Evolutionary Electronics: Automatic Design of Electronic circuits and systems by genetic algorithms*, (Boca Raton, London, New York, Washington DC: CRC Press, 2002), pp. 40-41

[32] scanned from Zebelum, et. al., p. 27

[33] James E. Gentle, Wolfgang Härdle, Yuichi Mori, Computational Statistics: an introduction, (http://www.quantlet.com/mdstat/scripts/csa/html/csahtml.html)

[34] Carnegie Mellon AI group FAQ (http://www.cs.cmu.edu/Groups/AI/html/faqs/ai/genetic/part2/faq-doc-3.html)

[35] Lukas Sekanina, *Evolvable Components,* (Berlin, Heidelberg, NY: Springer, 2003), pp. 28 - 32

[36] *ibid*, p. 16

[37] Zebelum, et. al., p.4

[38] Garrison W. Greenwood, Andrew M. Tyrell, *Introduction to Evolvable Hardware: A practical guide for designing self-adaptive systems*, (Hoboken, NJ: IEEE Press, John Wiley and Sons, Inc., Publication, 2007),

pp. 10-12

[39] Sekanina, p. 49

[40] Miller and Thomson, *Cartesian Genetic Programming,* (Berlin, Heidelberg: Springer, 2000), abstract

[41] Sekanina, pp. 42-47

[42] Greenwood and Tyrell, pp. 46-49

[43] Zebulum, et. al., pp. 9-10

[44] ibid., pp. 225-232

[45] Thompson, Layzell, Zebulum, http://www.cogs.susx.ac.uk/users/adrianth/TEC99/node17.html#cct_diag_fig

[46] Weiner, p.27

[47] http://www.custance.org/old/time/3ch1.html

[48] William Shakespeare, *Macbeth*

[49] *Oxford English Dictionary*

[50] *ibid.*

[51] Weiner, p. 89

[52] Weiner, p. 90

[53] Ludwig Wittgenstein, *Philosophical Investigations*, translated by G.E.M. Anscombe, (New Jersey: Prentice Hall, 1958), #115, p. 49e

[54] ibid, #66 and 67, pp. 31e-32e

[55] Margaret A. Boden, *The Creative Mind: Myths and Mechanisms*, (London: Routledge Press, 1990), pp. 43-51

[56] Yu-Tung Liu, "*What" and "Where" is design creativity: a cognitive model for the emergence of creative design*, (Idater, 96, Loughborough University)

[57] Boden, p. 21

[58] Christos, xi

[59] http://mathworld.wolfram.com/Attractor.html

[60] Wills, Lever, et. al., Attractor *Dynamics in the Hippocampal Representation of the Local Environment*, (Science Vol. 308. no. 5723, May 2005), pp. 873 – 876

[61] Elizabeth Svoboda, *Faces, Faces Everywhere*, (New York Times, February 13, 2007) , http://www.nytimes.com/2007/02/13/health/psychology/13face.html?ex=1177560000&en=1083f28c77a9f263&ei=5070

8.0 Annotated Bibliography

http://www.deweyhagborg.com/spurious/biblio.htm