INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006
Dubrovnik - Croatia, May 15 - 18, 2006.

DESIGN 2006

A SELF-TRAINING SYSTEM THAT LEARNS THROUGH EXPERIMENTATION

S. C. Braun and J. S. Gero

Keywords: autonomous agents, adaptive tools, machine learning

1. Motivation

The knowledge in computer programs is usually coded directly in the software or stored in a linked database. The program's developer provides the system with the knowledge he, or his client, thinks is needed to fulfil the task it is designed for. Commonly the developer gives the user the possibility to add more knowledge to the system, for example by adding a new dataset to the linked database. The system's knowledge can be described as predefined and fixed as it is not able to change without external help. That means the system is depended on the developer's or user's knowledge.
For some applications it is not possible or reasonable to predefine a system's coded knowledge. This is the case if the knowledge a system needs to fulfil a certain task is either unknown at the time of its development or too manifold and complex. Often developers code additional knowledge in case it may be useful.
In contrast to this kind of software, certain tools are able to adapt themselves to their task. Such tools are programmed to increase their knowledge through training. The training can be supervised by a user or the system can learn autonomously. In both cases training data has to be generated before the training phase begins. Usually this process needs to be carried out by a human user and takes time. Not only the design but also the training of the learning software is time consuming. Flexible, adaptive tools could be more useful and user-friendly if it were possible to reduce the time the user needs to expend on the training.

2. Aim

This paper presents an approach where such a system learns autonomously. The idea is to have a system that is able to generate new data and train itself with the knowledge derived from that data.
The exemplary system described here could form the basis of a tool that could be used to assist designers during the early stages of a product design. As any kind of design process is situated and dynamic [Gero and Kannengiesser 2004], systems that are developed to assist a designer during this process should be able to adjust to changing conditions and adapt themselves to their use. This paper presents both the conceptual basis and an implementation to demonstrate the ideas. The specific domain in the demonstration is the recognition and substitution of hand-drawn geometrical shapes. A tool like this could be used to transform a conceptual sketch into a refined CAD representation.

3. Related research

The concept of a learning adaptive system has been the subject of a large variety of approaches. Here we are inspired by elementary views of how humans learn. The current work is based on the ideas presented by Piaget [Piaget and Gruber 1995], who developed a model of human cognitive development. His theory is based on the notion that the developing child builds cognitive structures, in

other words, mental “maps”, schemes, or networked concepts for understanding and responding to physical experiences within his or her environment. Piaget stated that a child’s cognitive structure increases in sophistication with development, moving from a few innate reflexes such as crying and sucking to highly complex mental activities. Children can learn in many different ways: They learn from being told or being taught. "Being told" is giving an instruction or advice whereas "being taught" has to do with interaction. A teacher wants his pupils to understand what he is telling them. Another way for children to learn is to observe and imitate actions of other children or grown-ups. They also learn by experimentation or self-study, for example an infant learns, after several attempts, to correctly place a triangular brick into the triangular and not the square hole.
Just as there are different ways for children to learn there are different ways for a computer to be programmed to learn. You can supply computers with knowledge by storing knowledge in a database which can be accessed or by implementing instructions on how to react to a certain value of an input variable. This can be compared to a child "being told" what to do but, in the case of computers, this is not learning, rather it is data acquisition. It is possible to create structures, neural networks for example, that arrange themselves in accordance with what they have learned. There can be an interaction with a user during this process in which case the learning is supervised. If there is no interaction then it is unsupervised.
For a child, the different types of learning can be assigned to a particular stage of the child’s development, meaning the method of teaching which is most likely to be successful at a certain age can vary. Similarly, what kind of machine learning is the most promising at each stage of the development of a computer system and how can these learning processes be linked in order to build on the experiences achieved at a prior stage?

4. The Self-Training System

We will use the task of geometrical shape recognition as the domain in which we demonstrate the application of these thoughts. Conventional pattern recognition systems, based on neural networks with supervised learning, have to be trained by a human user during a specific training phase. The training data contains examples of inputs together with the corresponding outputs and the network learns to infer the relationship between the two [Pandya and Macy 1995].
The core of the system described below is a supervised learning neural network. What makes it different is that it creates its own training set and uses it for the training of its recognition engine. The system can be described as “self-training”. The advantage of this approach is that the user does not have to spend time on the generation of training data sets. The system still has to be provided with some basic knowledge and a basic structure on which to build, but it is flexible and dynamic enough to develop application specific and domain specific knowledge by itself.
A novel training set is produced through "experimental learning" as a curious agent [Saunders and Gero 2002] searches the combinations of known shapes, initially given to the system by a human user, for novel emergent shapes. As a result the system learns to recognize not only hand drawn representations of the shapes it has been taught but can also categorize shapes that it found independently through its own experiments with these initial shapes.

4.1 Method

The system’s recognition engine consists of a Multilayer-Perceptron (MLP), Figure 1. A MLP is a supervised learning neural network. In supervised learning training is accomplished by sending a given set of inputs through the network and comparing the results with a set of target outputs [Pandya and Macy. 1995].
The experiment must generate different representations of a sketched input shape as well as the target output for this shape. The experiment is carried out by a curious agent. An agent is an encapsulated computational system that is in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives [Jennings 2000].

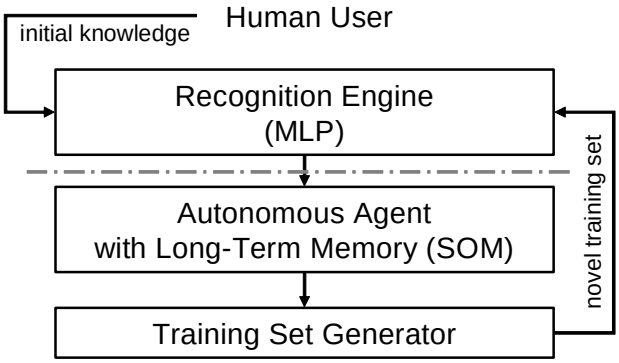

Figure 1. The system's components

The curious agent is composed of six primary functions: sensing, learning, detecting novelty, determining interest, planning and acting. In addition, the agent requires a long-term memory to store category prototypes [Saunders and Gero 2002]. Sensing samples the world to produce a stimulus pattern that characterises its environment according to the abilities of the agent. Learning updates prototypes stored in long-term memory to better reflect the agent's experiences as new types of stimulus pattern are produced. The differences between a new stimulus pattern and the closest matching category prototype are used to calculate a measure of the novelty for the experience. A measure of the interestingness of the current situation is then calculated based on a psychobiological model of preference to arousing stimuli. The goals of the agent are then updated to reflect the agent's current interest in its surrounding and actions are produced to propel the agent in an appropriate direction [Saunders and Gero 2002].
In this particular case the agent senses a combined image of two randomly chosen known shapes. The image is searched for emergent shapes and the agent determines the novelty of every emergent shape. When a novel shape is found a training set for this new shape is created and sent to the recognition engine. The agent's long-term memory consists of a self-organizing map (SOM). A SOM, like a MLP is a neural network. Unlike the MLP, the SOM learns unsupervised [Kohonen 1995].

4.2 Implementation

The structure of the overall system, Figure 1, can be divided into two stages which were implemented separately and then integrated:

- Stage 1: A neural network with supervised learning taught by a human user and a curious agent.
- Stage 2: A curious agent generating interesting emergent shapes in combination with a module that creates training samples of these interesting emergent shapes.

The following subsections describe these two stages and their implementation and the way they were integrated into one overall system.

4.2.1 Stage 1: A Neural Network for Pattern Recognition

The three major approaches for designing a pattern recognition system are (1) statistical, (2) syntactical or structural, and (3) artificial neural networks. This system uses a neural network to recognize hand drawn sketches of geometrical shapes.
The user decides which shapes the network should initially learn by providing both the inputs and the desired outputs. The input comprises several sketches (pixel graphics) of the same geometrical shape. The corresponding output consists of the refined shapes, Figure 2. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs iteratively as the weights are continually modified. During the training of a network the same

set of data is processed iteratively as the connection weights stabilise. This training procedure enables the system to categorize sketched representations of the shapes learned, for example triangles and rectangles, and replace them with the appropriate refined shapes.

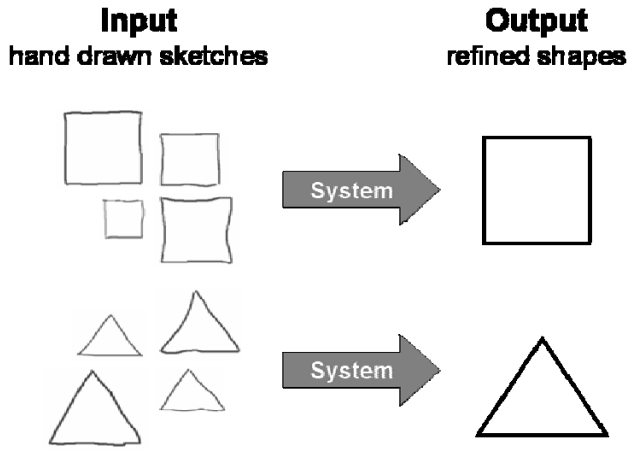

Figure 2. Stage 1: Supervised Learning – The system learns to infer the relationship between the input and the corresponding output

4.2.2 Stage 2: A Curious Agent

The second stage aims at generating new sketches, together with the corresponding shapes, which then can be learned by the agent. It consists of a module that chooses and combines sketches and shapes from the neural network's repertoire, a curious agent that searches the resultant combinations for interesting emergent shapes and a module that generates sketched training samples of the novel shapes. Figure 3 shows the different steps of this stage. In Figure 3 two squares are randomly chosen to be combined. The agent then searches for interesting emergent shapes. This process is modelled on the "Reflect-a-Sketch" system developed by Saunders [Saunders 2002]. The emergent shapes are isolated from their composite and their novelty assessed. The differences between a new shape and the closest matching category prototype are used to calculate a measure of the novelty for this shape. In the example in Figure 3 the agent chooses the bottom-left L-shape, step 2. This is a new geometry that has to be learned by the Multilayer-Perceptron, so that it will be able to recognize and replace a hand drawn L-shape, step 4 in Figure 3. In order to train the network new data has to be generated, step 3. This is done by a module which uses the sketched representation of the two combined squares as input as well as the information which of the emergent shapes has been judged interesting and novel by the "Reflect-a-Sketch" agent. The interesting shape is isolated from the rest of the sketch and is saved as "training data". As the reliability of the categorization increases with the number of representative training examples more sketched representations of the shape have to be generated and saved. The test data is sketched by a human and varies in size, proportion and precision. To achieve adequate classification results it is necessary to create slightly different representations of the shape. To obtain similar, yet different, representations of the sketched emergent shape the training data generator has to be provided with other representations of the initial input shape, in this case the square. These sketches can be taken from the original training set and need to be combined in the same way as the refined and sketched representations. This guarantees that the identical shape is isolated and saved. The system carries on generating new interesting, emergent shapes by randomly merging two shapes of its growing repertoire until a predefined number of different shapes is found. After the training phase is finished the system is able to recognize and replace the shapes initially given to it by the user as well as the shapes it learned through the experiments with the combinations of these shapes.

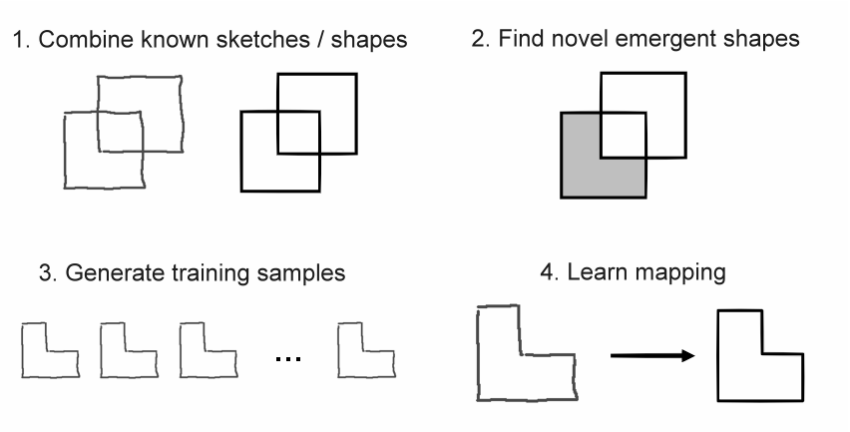

Figure 3. Stage 2: "Reflect-A-Sketch" algorithm and generation of a new training set

5. Tests

The idea was to create an adaptive system that, inspired by the diversity of human cognitive development processes, uses different kinds of machine learning to develop its expertise. The system briefly described here, combines a supervised learning recognition engine with an autonomous learning agent. The dependency of the system's learning success on the knowledge initially provided and the way it processes this knowledge is tested. The tests focus on the effects a variation of the initial repertoire has on the training set created. Four parameters that have major influence on the results have been studied:

1. The initial repertoire can consist of either three rectangular or three closed shapes. Three rectangles with different proportions have been chosen to represent the first category. For the second category the initial repertoire consists of a square, a triangle and a circle.
2. The repertoire from which the shapes for the combined images are taken can either stay the same throughout the experiment or grow. Grow means that every novel shape that is found is added to the repertoire and can then be used to create a new combined image.
3. In the case of a growing repertoire it makes a difference if the system "remembers" the shapes initially given to it or not. That means, if the system is programmed to "forget" its initial knowledge it will not use the three shapes it started the experiment with, after it found two novel shapes. Instead, from that point on it will produce combined images by joining only novel shapes.
4. Another tested restriction is to limit the growing repertoire to ten shapes. After the system found seven novel shapes it starts to replace existing shapes and in so doing eliminates the initial shapes.

The reasonable combination of the above parameters results in ten test cases.

6. Results and Conclusions

The results of the tests show that if the initial repertoire consists of only differently proportioned rectangles then the novelty value of the resulting emergent shapes is much lower than if the repertoire is composed of a triangle, a square and a circle. The combination of shapes that are more diverse leads to the discovery of more interesting shapes in these experiments.

The loss of the initial knowledge after the discovery of two novel shapes results in the generation of emergent shapes with a very low novelty value compared to those arising from experiments where the initial repertoire does not get discarded.

There are two conclusions that can be drawn from these results concerning behaviour that is very loosely based on human behaviour:

- The more diverse a system's initial knowledge, the more it can make out of it.

- A system needs initial knowledge to build on. If it forgets this basis it does not have any chance to create or learn something as interesting as a system that is able to remember its basic knowledge.

Here the successful combination of two types of machine learning has been presented. For the future, it is imaginable to create adaptive systems that integrate more than two different learning algorithms.

Acknowledgements

The research described in this paper is part of a diploma thesis which, thanks to the kind support of Prof. Udo Lindemann (TU Muenchen), was realized under the supervision of Prof. John Gero at the Key Centre of Design Computing and Cognition, Sydney, Australia.

References

Gero, J. S., Kannengiesser U., "The situated function–behaviour–structure framework", Design Studies, Vol.25, No.4., 2004, pp 373-391.

Jennings, N. R., "On agent-based software engineering", Artifcial Intelligence 117, 2000, pp 277-296.

Kohonen, T., "Self-organizing maps", Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 1995.

Pandya, A. P., Macy, R. B., "Pattern Recognition with Neural Networks in C++", CRC Press, 1995.

Piaget, J., Gruber, H., "The essential Piaget (100th Anniversary Edition)", Jason Aronson New York, 1995.

Saunders, R., "Curious Design Agents and Artificial Creativity: A Synthetic Approach to the Study of Creative behaviour", PhD thesis, Department of Architectural and Design Science, Faculty of Architecture, University of Sydney, February 2002.

Saunders, R., Gero, J. S., "Curious agents and situated design evaluations", in JS Gero and F Brazier (eds), Agents in Design 2002, KCDC, University of Sydney, Australia, pp. 133-149.

Stefanie Braun, Dipl.-Ing.
research assistant
Technische Universität München
Institute for Product Development
Boltzmannstrasse 15
D-85748 Garching
Tel.: +49 89 289 15126
Fax.: +49 89 289 15144
Email: stefanie.braun@pe.mw.tum.de
URL: http://www.pe.mw.tum.de