Creation By Refinement: A Creativity Paradigm for Gradient Descent Learning Networks

J. P. Lewis
Digital Sound Laboratory
New York Institute of Technology
Old Westbury, NY 11568

ABSTRACT

We describe a paradigm for creating novel examples from the class of patterns recognized by a trained gradient descent associative learning network. The paradigm consists of a learning phase, in which the network learns to identify patterns of the desired class, followed by a simple synthesis algorithm, in which a haphazard 'creation' is refined by a gradient descent search complementary to the one used in learning. This paradigm is alternative to one in which novel patterns are obtained by applying novel inputs to a learned mapping, and can be used for creative problems such as music composition which are not described by an input-output mapping. A simple simulation is shown in which a back propagation network learns to judge simple patterns representing musical motifs, and then creates similar motifs.

INTRODUCTION

The advantages which connectionist or neural network approaches have shown in other applications are potentially relevant to applications including simulation and computer arts which require the generation of novel patterns having a desired structure. For example, in simulation problems where existing models are inadequate for simulation, the simulation may be developed directly from samples of the data to be modeled.

The connectionist approach is particularly appropriate for computer arts applications such as *machine composition* of music [1,2] where the structure of the desired patterns is perceptually limited rather than determined by physical law in a more direct form. The problem of generating patterns constrained by this structure is somewhat parallel to the perceptual problems for which connectionist approaches are well suited.

Conversely, attempts to formulate satisfactory "laws" of composition (for example) have met with the difficulty that these laws are characteristically fuzzy and ill suited for algorithmic description. For example, in western tonal music a composition is considered to have a fundamental tone (tonic) which is understood throughout a composition and which should appear explicitly in the ending. In some cases a composition does not end on the tonic however, and occasionally a composition can be understood in terms of more than one tonic. Significantly, the existence of exceptions does not invalidate the notion of tonality; music exhibiting these exceptions may nevertheless be considered 'tonal' although we are unable to rigorously define what is meant by this.

We will consider several approaches to generating novel patterns with neural networks, and describe one approach, termed 'creation by refinement' (CBR), which is suited for non-representational creative problems such as music composition.

ALGORITHMIC APPROACHES TO CREATIVITY

Algorithms producing ostensibly novel output generally appeal in some way to a random number generator as the ultimate source of unpredictability. This statement brings up philosophical issues, perhaps the most interesting of which (from our viewpoint) is whether the complex 'emergent' behavior of a deterministic system could be considered novel. These issues have been debated extensively elsewhere (e.g. [3]); we will sidestep them by adopting the term 'artificial creativity' to denote machine generation of ostensibly novel patterns by appeal to randomness (the term is somewhat grandiose, but probably no more so than artificial intelligence is or was until recently).

Several algorithmic approaches to artificial creativity may be identified. In a *creation by filtering* or stochastic process approach, independent random numbers are filtered to impart a desired joint probability distribution (or moments thereof). In a *creation by perturbation* approach, the parameters of a deterministic system are perturbed by noise to generate novel outputs. In the absence of a distinction between a 'filter' and a deterministic system the distinction between these approaches is more a matter of viewpoint than definition.

NEURAL NETWORK APPROACHES TO CREATIVITY

A neural network approach reminiscent of 'creation by filtering', is to produce novel output by presenting novel input to a network trained with a desired mapping. This capability has been demonstrated, for example, in [4]. The evaluation of this approach as an artificial creativity paradigm in fact depends on the nature of the inputs. In the case of structured inputs (anything other than a random string), creative credit should be assigned to the act of generating the inputs rather than to the transformation implemented by the network, and the action of the network is properly called generalization (as in [4]) rather than artificial creativity.

The creation by filtering paradigm calls for strictly unstructured input. In this case however, supervised learning schemes encounter the following dilemma: For the network to have a soluble learning task, the random input vectors and the desired (training) patterns must be related by some fixed transformation. In many applications this transformation will not be known however, and in fact the motivation for adopting a neural network approach is to discover this transformation.

A second approach, reminiscent of 'creation by perturbation', is to generate novel behavior by randomly perturbing the weights in a trained network. The difficulty with this approach is that it is being creative with the problem rather than providing a 'creative solution': the weights in a trained network encode the structure of the problem domain, and so should not be significantly modified, whereas the variations in network state allowed by a given set of weights are not explored.

CREATION BY REFINEMENT

In this paradigm, a supervised learning algorithm is first trained to judge patterns from the space of possible creations. Sample 'creations' are presented at the input of the network, and a corresponding evaluation is provided as the judgement input (desired output).

Following training the inverse of the judgement function is probabilistically explored by the following procedure: The judgement input is set to a value representing a desirable creation, and the creation input is initialized to a random point within the creation space. The creation is then refined by a gradient descent search minimizing the error in judgement with respect to the creation.

This procedure is applicable for all supervised gradient descent learning approaches, since knowledge of the error gradient with respect to the weights entails being able to calculate this gradient with respect to the inputs: inputs may be considered as weights on 'virtual inputs' having a constant value. In the back propagation algorithm [5,6], the error gradient with respect to the inputs is available following the back propagation pass. Using the notation in [5], this is:

$$\frac{\partial E}{\partial i_m} = -\sum_n w_{m,n}\, \delta_n$$

where δ is the back-propagated error component, i_m is a component of the creation input, and the summation is over units receiving input from i_m.

We now consider some potential difficulties with this paradigm. The learning task has whatever difficulties are associated with the particular learning algorithm (local minima; the network configuration required to learn a particular judgement function may not be known in advance). The refinement procedure may also arrive at a local minimum which represents an unsatisfactory creation. Unlike most optimization applications however, (but in common with supervised learning applications) the magnitude of the error indicates the quality of a minimum, and so "we know if we're stuck"-- the minimization can be restarted from a different point if the minimum is poor.

In order for there to be more than one creation, the judgement function learned in training must be many-to-one. In fact the preimage of a particular judgement is potentially quite large. In the standard back propagation architecture, the space of equivalent inputs accepted by a particular network unit satisfies $\sum w_k\, i_k = c$ for a given constant c. This space is a hyperplane of dimension $n-1$ for a unit with n inputs. Similarly holding constant the outputs of all other units with these same inputs reduces the equivalent input space to an intersection of hyperplanes or affine

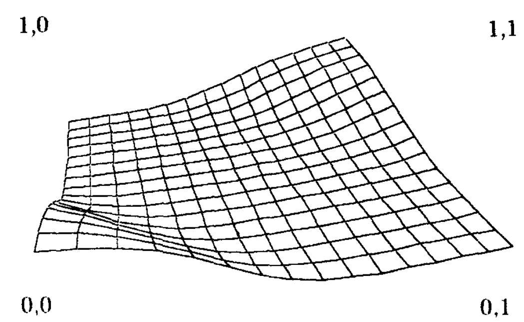


Fig. 1.
Error as a function of input for a back propagation network trained on the exclusive-or problem, and expecting a TRUE input relation.

set of typically much lower dimension, but the arbitrarily fixed outputs represent only one point in a hyperplane of inputs accepted by a subsequent network layer.

The preimage of a particular judgement must also be limited to acceptable creations. This requires that the training set adequately samples the input space; it must include counterexamples as well as examples. For example, in a back propagation network with two hidden units trained on the exclusive-or relation, the preimage of TRUE is approximately the the diagonal $x+y=1$ (Fig. 1). The preimage can be restricted to the desired points 0,1 and 1,0 by including the additional relation 0.5,0.5 $\rightarrow$ 0.5 in the training set. Since in general we do not know how to construct a training set which "adequately samples" the creation space, the training procedure should be amended to include the possibility of adding any undesirable creations to the training set and retraining.

For most applications the training judgements must be subjectively determined since objective knowledge of the judgement function will not be available. It is possible that errors in this determination could lead to an unstructured learning task in which itemization is the most economical description of the training pairs; in this case exploring the inverse of the judgement function will not yield new patterns with the desired structure. Refining or expanding the training set may solve this problem if it is recognized.

SIMULATION

To illustrate CBR, we chose a toy problem from tonal music composition, specifically, to generate short melodic figures which are "well formed" in a rudimentary way. The training data consisted of thirty manually generated five-note melodic figures, paired with a corresponding judgement of whether the figure was well formed or not (Fig. 2). Figures consisting mostly of intervals of unison (repetition), one scale degree (stepwise motion), thirds and fifths were considered well formed, while figures containing either excessive motion or excessive repetition were considered poorly formed (reflecting the notion that a good melody often describes a curve). Most of the well formed figures also used a common beginning and ending note.

Notes were encoded for a back propagation network using an itemization scheme like that used in NETtalk [7] to encode letters and phonemes, rather than by analog activation value. Each note was represented by a group of seven network inputs, with the strongest input encoding the note value. The network contained two layers of hidden units with 105 and 35 units respectively, and one output (judgement) unit.

The network learned to perform acceptably (maximum absolute error of 0.2) on the training set after about 4,700 presentations. The CBR procedure was then used to synthesize a number of similar figures (Fig. 2). The synthesized figures show that the network did learn a preference for stepwise and triadic motion, though the preference for beginning and ending on the same note was not learned.

Unfortunately this example does not demonstrate anything that could not have been achieved by an algorithmic (e.g. Markov chain [8]) compositional approach, though of course it was achieved without programming. We are exploring some slightly more realistic versions of the problem of composing melodies, although the present simulation suggests that serious musical composition incorporating chromatic scales, meter and rhythm, and harmony is well beyond the practical capabilities of a back propagation network implemented on most serial computers. (For comparison, the network used in this example contained about 7,300 weights versus about 10,000 reported in NETtalk [7]; we do not know if a smaller network could be used for this problem).

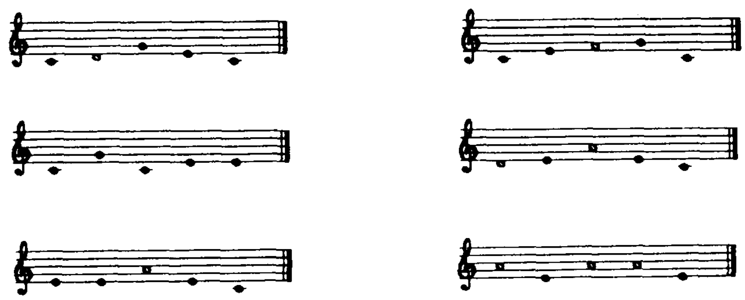
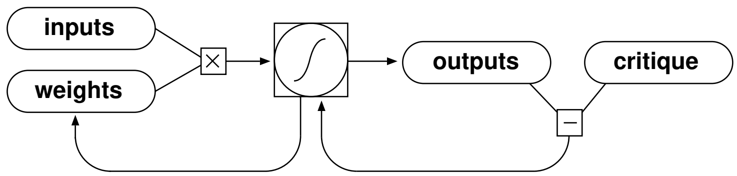


Fig. 2.
Samples of "well formed" melodic figures used in training (left) and figures generated by creation by refinement (right).

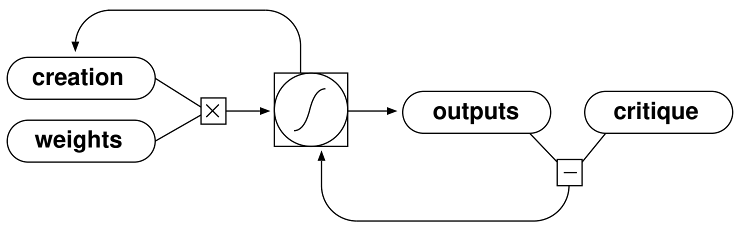
Despite the limitations of this example, we feel that 'creation by refinement' provides a fairly satisfactory approach to the generation of structured novel patterns, in particular for applications where the structure is perceptually determined. Considered as a paradigm for algorithmic creativity however, CBR is imitative at best since the generated patterns are limited to the structure discovered in the learning phase. In some ways it appears more creative to invent problems (or styles, in the case of art) than to solve them.

References

1. J. A. Moorer, Music and computer composition. *Communications of the ACM.* 15, 2 (1972), 104-113.
2. I. Xenakis, *Formalized Music.* Indiana University Press, Bloomington, Indiana, 1971.
3. J. Reichardt, Ed., *Cybernetics, Art, and Ideas.* New York Graphic Society, Greenwich, Conn., 1971.
4. T. Kohonen, P. Lehtio, and E. Oja, Storage and processing of information in distributed associative memory systems. in Parallel Models of Associative Memory (G. Hinton and J.A. Anderson Eds.), Erlbaum Assoc., Hillsdale, NJ, 1981, p. 105.
5. D.E. Rumelhart and J.L. McClelland, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* MIT Press, Cambridge, Mass., 1986.
6. D.B. Parker, *Learning-Logic, TR-47.* Center for Computational Research in Economics and Management Science, MIT, 1985.
7. T.J. Sejnowski and C.R. Rosenberg, *NETtalk: A Parallel Network that Learns to Read Aloud.* Johns Hopkins EECS technical report EECS-86/01, Baltimore, 1986.
8. K. Jones, Compositional applications of stochastic processes. *Computer Music Journal.* 5, 2 (1981), 45-61.



CBR training phase



CBR creation phase