Brock University

Department of Computer Science

Search Space Analysis of Recurrent Spiking and Continuous-time Neural Networks

Mario Ventresca and Beatrice Ombuki
Technical Report # CS-06-02
February 2006

Brock University
Department of Computer Science
St. Catharines, Ontario
Canada L2S 3A1
www.cosc.brocku.ca

Search Space Analysis of Recurrent Spiking and Continuous-time Neural Networks

Mario Ventresca and Beatrice Ombuki

***Abstract*—The problem of designing recurrent continuous-time and spiking neural networks is NP-Hard. A common practice is to utilize stochastic searches, such as evolutionary algorithms, to automatically construct acceptable networks. The outcome of the stochastic search is related to its ability to navigate the search space of neural networks and discover those of high quality. In this paper we investigate the search space associated with designing the above recurrent neural networks in order to differentiate which network should be easier to automatically design via a stochastic search. Our investigation utilizes two popular dynamic systems problems; (1) the Henon map and (2) the inverted pendulum as a benchmark.**

I. Introduction

THE search difficulty associated with a problem is tightly coupled with properties of its corresponding search space, commonly referred to as a fitness landscape. Although initially introduced by Sewall Wright [1] in the 1930's as a non-mathematical method for visualizing the effects of evolution, recent statistical and information theoretic procedures have been proposed which allows us to analyze a search space. These measures are capable of yielding enough information to give a general idea of the structure of the landscape. Depending on what these measures yield, it is possible to hypothesize whether the problem will be easy or difficult for a stochastic search.

To date many abstract problem search spaces have been analyzed, such as the Linear Ordering Problem [2]. More recently, the landscapes of more complicated real-world problems such as antenna design [3] and multi-legged robot neuro-controller [4] evolution have also been studied. The importance of conducting these investigations is mainly to be able to classify the expected behavior of a stochastic search a priori. Consequently, it also becomes possible to distinguish between approaches on another basis than simply the outcome of the algorithm. Also, this aids in determining when a particular stochastic search is more appropriate for a specific neural network model or task. The results of a search space analysis can also be used to aid in the development of search operators, such as crossover or mutation for evolutionary-based searches.

We present a landscape analysis of two rather different neural network architectures, both of which are utilized for similar time-dependant tasks. The first network type is classified by Mass [5] as being a member of the second generation of neuron (continuous-time), and the other is a more biologically accurate representation of neuron communication and classified as a third generation model (spiking). Our motivation for this study is to determine whether one of the network models is more amenable to stochastic search than the other. We have selected two common benchmark problems with chaotic behavior as a basis for our comparison; they are the chaotic time-series problem known as the Henon map, as well as control of an inverted pendulum.

The remainder of this article is organized as follows: Section II will briefly describe the problems under consideration followed by a formal description of fitness landscapes and the measures we utilized to conduct our investigation in Section III. In Section IV we will describe continuous-time and spiking recurrent neural networks. Section V will discuss our experimental results and setup. Finally, we will provide conclusions and future work in Section VI.

II. Problems Under Consideration

As mentioned above, we utilize the Henon map and inverted pendulum dynamic systems problems. This choice has been made because we are investigating recurrent neural networks, which are most commonly used for time-dependant problems [8].

A. The Henon Map

The Henon map is a prototypical two dimensional invertible map exhibiting chaotic solutions in the form of strange attractors [6]. Equations (1) and (2), represent the mathematical equivalent of repeatedly stretching and folding over and over again [6]. An initial point of the plane will either approach the set of points that defined the Henon strange attractor or diverge to infinity.

$$x(n+1) = 1 + y(n) - ax(n)^2 \tag{1}$$

This work was supported in part by the Natural Sciences and Engineering Council of Canada, NSERC.

M. Ventresca, was with the Department of Computing and Information Science, University of Guelph, Guelph, ONT N1G 2W1 CANADA. (M. Ventresca is currently a PhD student with Systems Design Engineering and member of the Pattern Analysis and Machine Intelligence Laboratory, at the University of Waterloo, Waterloo, ONT N2L 3G1 CANADA (e-mail: mventres@pami.uwaterloo.ca).

B. Ombuki is with Brock University, Computer Science Department, St. Catharines, ONT, L2S 3A1 CANADA. (e-mail: bombuki@brocku.ca).

$$y(n+1) = bx(n) \tag{2}$$

By plotting these deterministic equations over time we arrive at the Poincare plot presented in Fig. 1 (assuming common values of a = 1.4 and b = 0.3 [6]). The map's orbit traces out its characteristic banana-shaped strange attractor.

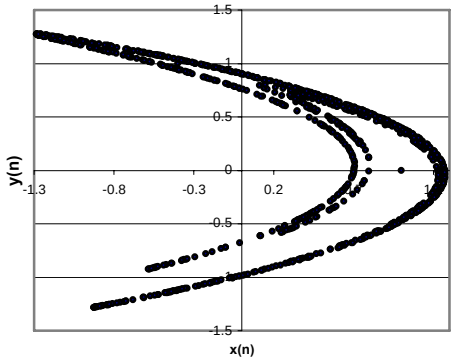

Fig. 1 The Henon strange attractor where a=1.4 and b=0.3.

B. The Inverted Pendulum

The inverted pendulum or cart-and-pole problem is a classic control theory benchmark problem that has been employed for more than 30 years [7]. It is a suitable test problem for neuro-controllers due to its high non-linearities and lack of stability [8]. This system is comprised of an inverted pole hinged on a cart that can move freely in the horizontal x-plane. The objective is to apply force to the cart at regular intervals such that the pendulum maintains an angle within 12 degrees of vertical and the cart lies within 2.4 units from the plane defined by x = 0. The state of the pendulum is shown in Fig. 2 and is defined by the cart position x, angle θ, horizontal velocity $\partial x/\partial t$, and angle velocity $\partial\theta/\partial t$.

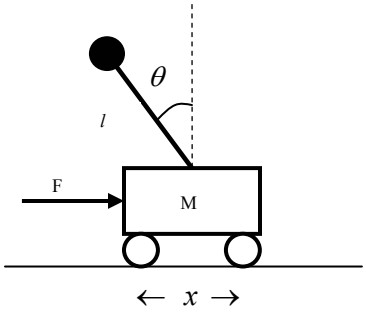

Fig. 2 The inverted pendulum system.

The state equations (3) and (4) are maintained by utilizing a 4th order Runge-Kutta approximation method [9], where m_p represents the mass of the pole, m is the mass of the pole and cart, l is the length of the pole, g is gravity and taken to be $9.81 ms^{-2}$ and F is the applied force on the cart.

$$\ddot{\theta} = \frac{mg\sin\theta - \cos\theta(F + m_p l\dot{\theta}^2 \sin\theta)}{l(\frac{4}{3}m - m_p\cos^2\theta)} \tag{3}$$

$$\ddot{x} = \frac{F + m_p l(\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta)}{m} \tag{4}$$

III. Fitness Landscapes

The fitness landscape concept was initially introduced as a non-mathematical aid to visualize the processes of selection and reproduction during evolution. The dimensionality of this landscape is defined by the number of dimensions in each solution representation plus a dimension representing its quality[1] or fitness. The structure of the landscape is therefore a function of both the representation and the search operators. The implication being that in order to facilitate efficient searching, operators should be designed in conjunction with the implicated structure of the fitness landscape. Therefore, if it is possible to construct landscapes which are easier to search, it is also likely that the search procedure will produce a higher quality solution than it otherwise would.

The structure of a fitness landscape is completely determined by the characteristics of smoothness, ruggedness and neutrality [10], which relate to differences in neighboring solutions quality. All three characteristics arise from the properties of the landscape's local optima. Assuming a maximization problem, a solution representation $g \in V$ is defined to be a *local maximum* if its fitness is greater than or equal to all of its neighbors, i.e., $f(g) \geq f(w) \forall\, w \in N(g)$, where the neighborhood *N(g)* is defined as the set of representations reachable by a single application of the search operator being considered [11]. A landscape is considered rugged if there is a high number of local optima present in the landscape.

In the event that few optima exist, the landscape may be either smooth or flat. If optima are characterized by large basins of attraction, the landscape is considered smooth. Assuming a maximization problem, a *basin of attraction* of a solution representation g_n is defined as the set of vertices $\mathrm{B}(g_n) = \{g_0 \in V \mid \exists g_1, ..., g_n \in V$ with $g_{i+1} \in N(g_n)$ and $f(g_{i+1}) > f(v_i) \forall\, i, 0 \leq i < n\}$ [11]. The size of a basin is generally considered to be defined as the number of representations within it. Those local optima with small attractive basins are called *isolated* [11].

Landscapes that are characterized by few local optima generally contain large amounts of *neutrality* which arises as movement along *neutral networks* (solutions of equal fitness but differing solution representation). These neutral areas of

[1] Where solution quality is problem dependant, for example sum-squared error is commonly used to gauge neural network quality.

a landscape are a result of the presence of plateaus and ridges. As defined by Jones [11], a *plateau* is a subset P of two or more solution representations such that for every pair of representations $g_0, g_n \in P$ a subset of representations $g_1,...,g_{n-1}$ exists where $f(g_i) = f(g_{i+1})$ and $g_{i+1} \in N(g_i) \forall\, 0 \le i < n$.

A. Statistical Landscape Measures

Early work by Weinberger [12] aimed to gain insight into the fitness landscape structure by examining successive fitness values of a random walk on the landscape. The walk is represented as a time-series and the measure is arrived at via an autocorrelation analysis of successive fitness values. The resulting value is a number between 0 and 1, where values further from 0 indicate the degree to which neighboring solution representations are correlated. The implication is that highly correlated landscapes are very smooth and thus should be relatively easy to search. Similarly, rugged landscapes are not very correlated and have many local optima, meaning that they are more difficult to search, although this is not always the case. This measure of landscape ruggedness is calculated via the empirical autocorrelation function shown in (5), where f_t is the fitness of the solution representation after t applications of the search operator, $\bar{f}$ is the average fitness of a walk of length T, the parameter i represents the time lag of the series.

$$r(i) = \frac{\frac{1}{T-i}\sum_{t=1}^{T-i}(f_t - \bar{f})\cdot(f_{t+1} - \bar{f})}{\frac{1}{T}\sum_{t=1}^{T}(f_t - \bar{f})^2} \tag{5}$$

Stadler [13] derived the *correlation length* as an estimate to approximate the number of applications of the search operator are required to cause a "descendant" representation to cease having a correlated fitness value with its "ancestor". This measure is based on the fact that empirical autocorrelation functions often show an exponential decay of the form $r(i) = r(1)^i = e^{-i/l}$, which when solved for the walk length l, results in the correlation length measure shown in (6).

$$l = -\frac{1}{\ln(|r(1)|)} \tag{6}$$

A high correlation length indicates a rather smooth landscape. Conversely, a low correlation length implies neighboring fitness values are not very similar, thus indicating a somewhat rugged landscape.

B. Information Theoretic Landscape Measures

The information theoretic counterpart to the aforementioned statistical approaches also approximates characteristics of the landscape via a random walk-based analysis [14]. However, this approach measures the amount of entropy or fitness change along the walk. We use the four measures proposed by Vassilev, Fogarty and Miller [14].

The *Information Content* (7) measures the ruggedness with respect to the flat or neutral areas of the landscape. The degree of flatness sensitivity of this measure is based on an empirically decided parameter ε, restricted to the range [0…L], where L is the maximum fitness difference along the random walk. Consequently, the analysis will be most sensitive when ε is equal to 0.

$$H(\varepsilon) = -\sum_{p \ne q} P_{[pq]} \log_6 P_{[pq]} \tag{7}$$

The probabilities $P_{[pq]}$ represent the frequencies of the possible fitness transitions from p to q while performing the random walk. Each of the *[pq]* blocks are elements of the string $S(\varepsilon) = s_1 s_2 s_3 ... s_n$, of symbols $s_i \in \{\bar{1}, 0, 1\}$, where each s_i is obtained for a particular value of ε based on equation (8), meaning that $s_i = \Psi_f(i, \varepsilon)$. Thus, ε essentially provides an accuracy or sensitivity of the landscape analysis.

$$\Psi_f(i,\varepsilon) = \begin{cases} \bar{1}, \text{ if } f_i - f_{i-1} < -\varepsilon \\ 0, \text{if } |f_i - f_{i-1}| \le \varepsilon \\ 1, \text{if } f_i - f_{i-1} > \varepsilon \end{cases} \tag{8}$$

The *Partial Information Content* (PIC) measure is used to indicate the modality or number of local optima present in the landscape. The idea behind this measure is to filter out non-essential parts of $S(\varepsilon)$ in order to acquire an indication of the modality of the random walk and therefore of the landscape. Equation (9) gives the formula to calculate the PIC, where n is the length of the original walk and μ is the length of the summarized string $S'(\varepsilon)$.

$$M(\varepsilon) = \frac{\mu}{n} \tag{9}$$

The value of μ is determined via the recursive function defined in Equation (10), and calculated using $\mu = \Phi_S(1, 0, 0)$.

$$\Phi_S(i,j,k)=\begin{cases} k & \text{if i > n} \\ \Phi_S(i+1,i,k+1) & \text{if j = 0 and } s_i \neq 0 \\ \Phi_S(i+1,i,k+1) & \text{if j > 0, } s_i \neq 0 \text{ and } s_i \neq s_j \\ \Phi_S(i+1,j,k) & \text{otherwise} \end{cases} \quad (10)$$

When the value of $M(\varepsilon)=0$ it indicates that no slopes were present on the path of the random walk, meaning that the landscape is rather flat. Similarly, if $M(\varepsilon)=1$ then the path is maximally multimodal. Furthermore, it is possible to calculate the expected number of optima of a random walk of length *n* via

$$E[M(\varepsilon)]=\left\lfloor \frac{nM(\varepsilon)}{2} \right\rfloor. \quad (11)$$

Related to the modality is *Information Stability* (ε^*) which is the smallest value of ε that results in the information content being equal to 0. High information stability indicates that the largest possible fitness difference between any two neighboring solution representations is relatively high, and therefore the landscape may be very steep and perhaps rugged.

The *Density-Basin Information* (DBI) measure given in equation (12), and indicates the flat and smooth areas of the landscape as well as the density and isolation of peaks in the landscape. Thus, it provides an idea of the landscape structure around the optima.

$$h(\varepsilon)=-\sum_{p\in\{\bar{1},0,1\}} P_{[pp]}\log_3 P_{[pp]} \quad (12)$$

where $P_{[pp]}$ represents the probability of sub-blocks 00, $11, \bar{1}\bar{1}$ occurring. A high number of peaks within a small area would result in a high DBI value. Conversely, if the peak is isolated the measure will yield a low value. Thus, this information gives an idea as to the size and nature of the basins of the landscape. Landscapes with a high DBI content should be easier for an evolutionary algorithm to attract to an area of fitter solutions. In contrast, it is likely that for landscapes with a low DBI value an evolutionary algorithm is less likely to discover regions of high fitness.

IV. Neural Network Models

A. Continuous-time

Continuous-time recurrent neural networks (CTRNNs) [15, 16] are composed of dynamical neurons. That is, they are constituted by neurons that tend to change their activation state at different rates according to a time constant parameter $\tau > 0$. Each neuron acts as a leaky integrator, so that input increases a neuron's action potential which then slowly degrades to a resting value over time. Thus, the neuron has the inherent nature of utilizing its previous state(s) in order to influence its current state. For CTRNNs this state is characterized by both the activation potential and the activity of the neuron, the latter being defined by the type of activation function used. Thus, the purpose of the time constant is to determine the rate of change of the activation potential. This process is summarized as

$$v_i=\frac{1}{\tau_i}\left(-v_i+\sum_{j=1}^{N} w_{ji}\sigma(v_j+\theta_j)+I_i\right) \quad (13)$$

where v_i is the activation potential of the i^{th} neuron and $\tau_i > 0$ is its time constant. The activation function σ is composed of each j^{th} incoming signal as well as a bias term θ_j and I_i represents a constant external input signal. The activation function used for this work is the common logistic function shown in (14).

$$\sigma(x)=\frac{1}{1+e^{-x}} \quad (14)$$

The dynamics of the differential equation are solved using the standard Euler method [9] with a step size of 1.

It has been shown by Funahashi and Nakamura [17] that CTRNNs are universal approximators of any real valued function**.**

B. Spiking

Also known as Pulsed Neural Networks [18], spiking neural networks (SNNs) attempt to utilize a more biologically realistic third generation neuron model [5]. It has been argued by Mass [18, 19] that SNNs are much better suited for applications where the timing of signals carries important information. Naturally, SNNs can also be applied to the same problems as non-spiking networks. These neural models do not imply that other non-spiking models are obsolete. However, it has been shown mathematically by Mass [20] that networks of spiking neurons have considerably more processing power than similarly sized non-spiking networks.

Although many spiking neural models exist we have chosen to implement the Spike Response model proposed by Gerstner [21]. Under this model a neuron is described solely by its membrane potential v_i at a given time *t*. When the membrane potential reaches the neuron's threshold level ϑ the neuron will fire, represented as $t_i^{(f)}$. The set of all firing times of neuron *i* (spike-train) is defined as

$$F_i=\left\{t_i^{(f)}:1\le f\le n\right\}=\{t\mid u_i(t)=\vartheta\}. \quad (15)$$

After emitting a spike, the neuron enters into a period of

absolute refractoriness, lasting for a period of Ω time units, during which it cannot emit a spike. After this period the neuron enters the relative refractory period where it becomes increasingly likely that it can emit a spike. It is common to see the relative refractory modeled according to the kernel equation shown in (16).

$$\eta(s) = \begin{cases} -\vartheta e^{\frac{-s}{\tau_m}} & \text{if s} > \Omega \\ -\infty & \text{otherwise} \end{cases} \quad (16)$$

where $s = (t - t^f)$ is the difference between the current time and the time the spike was evoked and τ_m represents the membrane time constant. A plot of the relative refractory period is shown in Fig. 3.

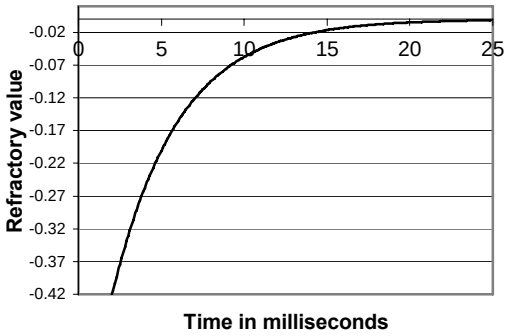

Fig. 3 Plot of (16) where θ = 0.7, τ_m = 4ms and Ω = 2ms.

Properties of the actual spike are a function of (1) the delay Δ between the firing time at the presynaptic neuron and the time it arrives at its post synaptic destination, (2) a synaptic time constant τ_s, and (3) a membrane constant τ_m. A common function used to represent a spike is shown in equation (17).

$$\varepsilon(s) = \begin{cases} e^{-\frac{s-\Delta}{\tau_m}}\left(1 - e^{-\frac{s-\Delta}{\tau_s}}\right) & \text{if s} \geq \Delta \\ 0 & \text{if x} < \Delta \end{cases} \quad (17)$$

Fig. 4 shows equation (17) where Δ = 1, τ_s = 10*ms* and τ_m = 4*ms*.

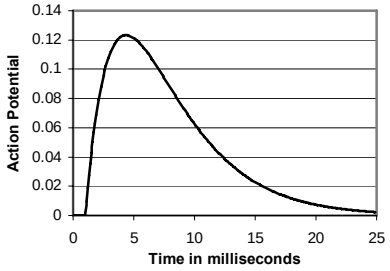

Fig. 4 An example of a function (17) describing the spike of a neuron.

By combining (16) and (17) we can describe the dynamics of a single neuron having several excitatory and inhibitory inputs. Each of the incoming signals is given a synaptic weight w_j whose sign is positive if the signal is excitatory or negative otherwise. The membrane potential of a neuron *i* is then described as

$$v_i(t) = \sum_j w_j^t \sum_f \varepsilon_j(s_j) + \sum_f \eta_i(s_i). \quad (18)$$

If this membrane potential is greater than or equal to the neuron's firing threshold ϑ, the neuron will emit a spike and η_i takes the value of $-\infty$, thus preventing an immediate new spike.

For this work each synapse may has a different synaptic time constant τ_s as well as a different delay Δ and weight *w* which affects the shape of the spike. Similarly, each neuron also has a different membrane time constant τ_m. We fix the threshold value ϑ (although we fix this value for all neurons).

V. Experimental Results And Discussion

Before we discuss the experimental results, it is necessary to first present some details regarding solution encoding and the search operator used.

A. Experimental Setup

We directly encode continuous-time neural networks into a set of three matrices representing each connection weight $w_{ij} \in [-5,5]$ between neurons *i* and *j*, each time constant $\tau_i \in [1,50]$ and each bias $\theta_i \in [-1,1]$. In a similar manner, spiking networks were encoded into a set of three matrices representing connections weights $w_{ij} \in [-5,5]$, each axonal delay $\Delta_{ij} \in [1,5]$ and the synaptic and membrane potentials $\tau_s, \tau_m \in [1,5]$, where $\tau_m > \tau_{sj}\ \forall j$.

Each random walk begins with a randomly generated neural network where the network parameters are bounded on the intervals defined above. The search operator works by selecting a set *Q* of neural parameters to modify. The probability of a parameter being selected is *1/k*, where *k* is the maximum number of that specific parameter. The search operator then adds a small random value $r \in U(-0.5, 0.5)$ to each $q \in Q$. This process is repeated for each step along the random walk.

We have conducted two sets of experiments for both the Henon map and inverted pendulum problems. The first involves empirically decided fixed neural architectures and the second allows for the removal or addition of connection weights with the probability 0.15 after being randomly selected, as described above. In either case, 1,000 random walks of length 100 were generated, thus each landscape analysis is based on 100,000 neural networks.

B. Henon Map

We construct SNNs and CTRNNs with two input, three hidden and two output neurons and evaluate them via 1,000 consecutive Henon values where we aim to minimize the sum-squared error (SSE) of the network. Input into the CTRNN network is initially *{0, 0}*, and for the spiking network we cause 10 equally spaced spikes over the first 10*ms* of simulation. We stop feeding data into the network and instead just record the output neurons' activation values. SNN outputs are transformed into raw values by $\{x/10-2, y/10-2\}$ where x and y are the first times a spike is emitted from each output neuron. We subtract two because spike times are positive values and Henon values can be negative, but less than 2. Also, we divide by 10 because given an input value the network has 10*ms* to process it. Furthermore, either network must extrapolate the next 10 chaotic Henon series points from only an initial point.

Table 1 shows the statistical landscape results of both fixed and adaptable networks. We can see that the average fitness of fixed architecture CTRNNs is relatively very high. However, a much lower best fitness was encountered (since the optimal solution is 0). It is also seen that the adaptable SNN fairs poorly compared to its fixed architecture counterpart. This is the reverse scenario for CTRNNs, and is likely due to the sensitivity of spiking networks.

TABLE 1
RESULTS OF STATISTICAL LANDSCAPE ANALYSIS

Network		Average Fitness	Best Fitness	Auto-correlation	Correlation length
CTRNN	Fixed	10,016.6	2,028.8	0.9499	19.4798
	Adapt	1,9929.2	690.7	0.7076	2.8916
SNN	Fixed	9,910.8	1,230.8	0.8233	5.1426
	Adapt	9,367.8	2,552.2	0.5181	1.5207

Since the autocorrelation values are closer to 1 (resulting in a high correlation length) for CTRNNs than SNNs, it is deduced that the CTRNN landscapes are smoother than those of the SNNs. It is then expected that the CTRNN search space is easier to explore. However, the very large correlation length for fixed architecture CTRNNs could also indicate the presence of large neutral areas, or of a very large peak.

TABLE 2
RESULTS OF INFORMATION LANDSCAPE ANALYSIS

Network		Info. Content	Density-basin Info.	Partial-info. content	Expected # of optima
CTRNN	Fixed	0.4186	0.6114	0.5333	26.0000
	Adapt	0.0126	0.0083	0.0249	1.1475
SNN	Fixed	0.4700	0.4707	0.2870	13.9020
	Adapt	0.0352	0.0929	0.0719	3.5064

The information-theoretic measures shown in Table 2 yield much more information about the structure of the fitness landscape than does the statistical analysis.

The information content is similarly quite large for both fixed architecture experiments, indicating the presence of a variety of shapes on the landscape. The remaining measures are not so similar (for fixed networks). From the relatively high density-basin information value we see that the CTRNN landscape contains diverse flat and smooth landscape sections. Furthermore, from the partial-information content and expected number of optima it is also found that the landscape contains a rather high degree of modality compared to the SNN landscape. Thus, both landscapes contain a mixture of smooth and rugged areas. So, depending on where we are currently searching, the characteristics of the landscape may vary drastically. Therefore, the starting point of a search may have a large influence on the outcome of the search. Furthermore, the SNN landscape seems to contain more flatness, but in order to confirm this hypothesis it is necessary to plot the measures against increasing ε. Since Fig. 5 shows monotonically decreasing landscape functions we conclude that the landscape flatness prevails over ruggedness.

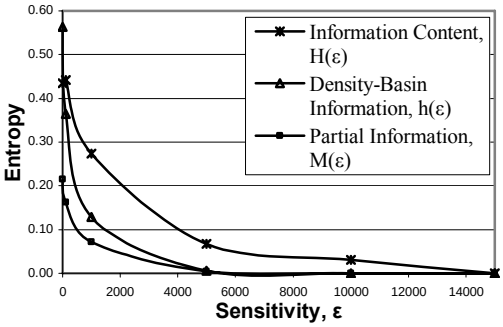

Fig. **5** Information measures as a function of sensitivity.

In a similar fashion we can plot the CTRNN values (although not done here), which leads to the observation that they are not monotonically decreasing and therefore the landscape indeed contains a relatively high degree of ruggedness.

The adaptive network landscape analysis yields a much different picture. All measures are drastically smaller than before which indicates the presence of a very high degree of smoothness. So, we can hypothesize that the landscape contains very few peaks and is very smooth. It could be difficult to discover a solution if most of the landscape is flat because most search time is spent traveling long low quality neutral networks. If the landscapes are very smooth then a high quality solution will likely be easy to discover because of the presence of a very wide peak. However, the low density-basin information (DBI) values suggest that the basins are relatively small and therefore it is likely that flatness prevails.

C. Inverted Pendulum

The goal of the neural networks is to control an inverted pendulum placed 0.5 units left of the *x=0* plane at an angle of 5^o for 400*s*. The fixed architecture neural networks were constructed with four input (2 for angular and 2 for horizontal movement), three hidden and two output neurons. We present the current angle and x-axis location to the networks. Two inputs are dedicated to each in order to distinguish between the left and right side of the x=0 plane. Furthermore, since the velocity and rate of change of the angle are not presented as inputs, an internal representation of the derivative of each value must be maintained within each neural network [8]. The output neuron with the highest value (or most spikes for SNN) causes a 10N force to be applied pushing the cart to the left or right, respectively. In order to determine the quality of a solution to this task we use the following function proposed by [8], which is to be maximized

$$P = \sum_{t=1}^{t_{\max}} \Delta\left(0.5\left(1-\frac{15\,|\,\theta(i)\,|}{\pi}\right)+0.5\left(1-\frac{|\,x(i)\,|}{2.4}\right)\right). \quad (19)$$

As for the Henon map, SNN inputs needed to be transformed into spike times. The transformation used causes the evocation of a spike from the appropriate input neuron for every 4^o from vertical or 0.5 units from the x=0 plane. Therefore the closer to disobeying the limit rules (described in Section II) the network gets, the more excited it becomes.

We can see from Table 3 that the spiking network greatly outperforms the continuous-time model in terms of fitness score. However, the auto-correlation and correlation length measures indicate that the spiking network landscape is much more rugged that its counterpart. The adaptable SNN showed a large improvement over the fixed architecture version, whereas there is no significant difference between fixed and adaptable CTRNN networks.

TABLE 3
RESULTS OF STATISTICAL LANDSCAPE ANALYSIS

Network		Average Fitness	Best Fitness	Auto-correlation	Correlation length
CTRNN	Fixed	0.1566	1.8303	0.9232	12.5185
	Adapt	0.1578	2.7913	0.9300	13.7818
SNN	Fixed	0.2892	3.7594	0.7878	4.1918
	Adapt	0.3317	8.5064	0.8364	5,5989

Table 4 shows the information-theoretic results. All of the CTRNN measures are very low, which indicates that the landscape contains a high degree of smoothness. The low partial-information content and expected optima signify that only few peaks can be found, whether they are isolated depends on the DBI value. Since the density-basin information is similarly low then the basins of attraction are small and the peaks are isolated.

In contrast, the SNN landscapes contain a larger variety of shapes and so are more rugged, as is suggested by the relatively high information and partial-information content values. Additionally, the density-basin measure also yielded relatively high values implying that the size of the attractive basins surrounding the peaks is also relatively large.

TABLE 4
RESULTS OF INFORMATION LANDSCAPE ANALYSIS

Network		Info. Content	Density-basin Info.	Partial-info. content	Expected # of optima
CTRNN	Fixed	0.0977	0.0395	0.0191	0.8940
	Adapt	0.0952	0.0388	0.0180	0.8530
SNN	Fixed	0.2600	0.1301	0.0766	3.5840
	Adapt	0.5043	0.3652	0.1929	9.3020

A plot of the SNN landscape indicators versus sensitivity is used to determine whether the landscape is more smooth than rugged. From Fig. 6 we can deduce that the landscape does contain flat areas. Furthermore, there is the existence of a large peak, which has a rather large basin surrounding it.

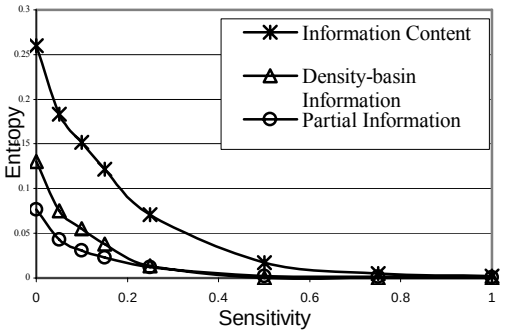

Fig. 6 Information measures as a function of sensitivity.

Therefore, the CTRNN landscape is much harder to search than that of the SNN. This is true for both fixed and adaptable neural architectures.

VI. CONCLUSION

We have presented a comprehensive search space analysis of continuous-time and spiking recurrent neural networks based both on statistical and information-theoretic measures. It was found that under the assumptions of network encoding and the search operator, the SNN search space is more amenable to stochastic search for control problems such as the inverted pendulum. However, when the problem requires precise output from the network, as was the case for the Henon map experiments the CTRNN network itself may be more practical. This limitation is due to the spike encoding requirement that is not present in the CTRNN model. Perhaps a different, computationally practical spike encoding might yield more competitively accurate results. This forms a possible direction for future work.

Another possible direction for future work involves analyzing different solution representations, as well as different search operators and combinations thereof. Additionally, various stochastic searches can be developed in order to confirm the hypotheses laid out in this work. Also, the metrics used here assume the landscape is isotropic, meaning the landscape is similar in structure throughout. Whether this is in fact true for neural networks is an open problem [22]. An investigation of local search spaces may lead to a greater understanding of the larger space, and perhaps lead to the development of other landscape measures.

REFERENCES

[1] S. Wright, "The Role of Mutation, Inbreeding, Crossbreeding and Selection in Evolution.", *Proceedings, Sixth International Conference on Genetics*, pp. 356-366, Oxford University Press, 1932.

[2] T. Schiavinotto and T. Stützle. "Search space analysis of the linear ordering problem" *G. Raidl et al., editor, Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2003*, number 2611 in Lecture Notes in Computer Science, pages 322-333. Springer Verlag, Berlin, Germany, 2003.

[3] J. T. Alander, L. A. Zinchenko, S. N. Sorokin. "Analysis of Fitness Landscape Properties for Evolutionary Antenna Design," *IEEE International Conference on Artificial Intelligence Systems (ICAIS'02)*, pp. 363-368, 2002.

[4] J. Teo and H. A. Abbass,., "Search Space Difficulty of Evolutionary Neuro-Controlled Legged Robots", *Proceedings, 6th Australian-Japan Joint Workshop on Intelligent and Evolutionary Systems*, pp. 54-61, 2002.

[5] W. Maas, "Networks of Spiking Neurons: The Third Generation of Neural Network Models", *Transactions of the Society for Computer Simulation International*, pp. 1659-1671, 1997.

[6] M. Hènon, "A Two Dimensional Mapping with a Strange Attractor", *Communications of Mathematical Physics*, vol. 50,pp. 69-78, 1976.

[7] W. T. Miller, R. S. Sutton, and P. J. Werbos,, *Challenging Control Problems, Neural Networks for Control*, MIT Press, 1990.

[8] F. Pasemann, "Pole-balancing with Different Evolved Neurocontrollers", *Proceedings, International Conference on Artificial Neural Networks*, pp. 823-829, Springer-Verlag, 1997.

[9] F. Diacu, *An Introduction to Differential Equations: Order and Chaos*, W.H. Freeman & Company, 2000.

[10] V. K. Vassilev, T. C. Fogarty, and J. F. Miller, "Fitness Landscapes: from Theory to Application", *Advances in Evolutionary Computation: Theory and Applications*, pp. 3-44, Springer, 2003.

[11] T. Jones, "Evolutionary Algorithms, Fitness Landscapes and Search", *PhD. Thesis*, University of New Mexico, 1995.

[12] E. Weinberger, "Correlated and Uncorrelated Landscapes and How to tell the Difference", *Biological Cybernetics*, vol. 63, pp. 325-336, 1990.

[13] P. Stadler, "Towards a Theory of Landscapes", *R.Lopéz-Peña, R.Capovilla, R.García-Pelayo, H.Waelbroeck, and F.Zertuche (eds.), Complex Systems and Binary Networks, Proceedings of the Guanajuato Lectures*, pp. 77-163, Springer-Verlag, 1995.

[14] V. K. Vassilev, T. C. Fogarty and J. F. Miller, "Information Characteristics and the Structure of Landscapes", *Evolutionary Computation*, vol. 8, no.1, pp. 31-60, 2000.

[15] R. Beer, "On the Dynamics of Small Continuous-time Recurrent Neural Networks", *Adaptive Behavior*, vol. 3, no.4, pp. 469-509, 1995.

[16] R. Beer and J. C. Gallagher, "Evolving Dynamical Neural Networks for Adaptive Behavior", *Adaptive Behavior*, vol. 1, no.1, pp. 91-122, 1992.

[17] K. Funahashi, and Y. Nakamura, "Approximation of Dynamical Systems by Continuous-Time Recurrent Neural Networks", *Neural Networks*, vol. 6, pp. 801-806, 1993.

[18] W. Maas and C. Bishop, *Pulsed Neural Networks*, MIT Press, 1999.

[19] W. Maas, "Computation with Spiking Neurons", *The Handbook of Brain Theory and Neural Networks, 2nd Edition,* pp. 1080-1083, MIT Press, 2001.

[20] W. Maas, "Lower Bounds for the Computational Power of Networks of Spiking Neurons", *Neural Computation*, vol. 8, no.1, pp. 1-40, 1996.

[21] W. Gerstner and W. Kistler, *Spiking Neural Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, 2002.

[22] T. Smith, P. Husbands, and M. O'Shea, "Not Measuring Evolvability: Initial Investigation of an Evolutionary Robotics Search Space", Proceedings, Congress on Evolutionary Computation vol. 1, pp. 9-16, 2001.