PHYSICAL REVIEW E VOLUME 54, NUMBER 5 NOVEMBER 1996

Frequency analysis with Hopfield encoding neurons

Epifanio Bagarinao, Jr. and Caesar Saloma*

National Institute of Physics, University of the Philippines, Diliman, Quezon City 1101, The Philippines

(Received 29 April 1996)

A three-layer network that utilizes Hopfield encoding neurons [J. J Hopfield, Nature **376**, 33 (1995)], is designed to compute the Fourier spectrum of an analog input signal of duration T. Each of the $2M$ (integer $M>0$) Hopfield neurons in the input layer, is linked to an exclusive decoder with its output connected to all the $4M$ neurons present in the output layer. The connection strength between a decoder and a target neuron in the output layer (hereby called the output neuron), is characterized by a coupling constant which attenuates the decoder output that reaches the output neuron. All the attenuated $2M$ decoder signals reaching an output neuron are summed up within an integration time given by the period of the oscillatory drive of the Hopfield neuron. The frequency resolution and bandwidth of the analyzer network are given by $1/T$ and $2M/T$, respectively. The first $2M$ output neurons yield the amplitudes of the real components of the Fourier spectrum, while the next $2M$ output neurons give the amplitudes of the corresponding imaginary components. Experiments show that the network exhibits an exponentially decaying learning error, and is capable of learning the general properties of Fourier transform from a limited set of examples. [S1063-651X(96)08510-8]

PACS number(s): 87.10.+e, 84.35.+i, 07.05.Mh

I. INTRODUCTION

Knowing how the nervous system represents external information is essential in our quest to understand both the relation between form and function in such systems [1,2]. A simple information coding scheme is often carried out by a signal detector with a simple architecture [3]. For biological networks, it has long been taken that information is encoded in the number of pulses that a sensing neuron generates per unit time in response to a particular stimulus value [4,5]. This amplitude-to-frequency data conversion scheme is inherently slow and may not be the only coding scheme employed by biological networks.

An alternative scheme is the temporal code representation which encodes specific information regarding the external stimulus in the relative distances of the spikes with respect to a fixed reference point. Temporal coding has received more attention lately [6,7] due to the recent model by Hopfield [8] of an encoding neuron with response characteristics that accommodate the observed behavior of certain biological networks [9–12]. Hopfield neurons utilize temporal coding to represent the value of the external stimulus $s(x)$ at a particular value of x. Variable x may represent space, time, etc.

The aim of this paper is to show that a network of Hopfield encoding neurons can learn general rules from only a limited set of examples. In particular, a three-layer network with its input layer formed by Hopfield neurons, is trained to compute the (complex) Fourier spectrum $S(f)$ of $s(x)$, where f is the frequency variable. The spectrum $S(f)$ describes the relative amplitudes and phases of the component sinusoids of $s(x)$.

Fourier analysis is a versatile signal processing tool [13–15] that can be used to enhance image contrast, highlight specific signal features like edges and ridges, and uncover nonlocal characteristics of a signal in the presence of noise and unwanted background. Edge detection, for example, maybe accomplished by suppressing the low-frequency component sinusoids of $s(x)$, while detection of $s(x)$ in the presence of noise, maybe accomplished by suppressing the high-frequency component sinusoids that contain most of the noise signal. The contrast of a barely discernible signal may be enhanced by proper selection of both the gain factor and the threshold level for the detectable component sinusoids.

Because the signal processing tasks being mentioned are also equally important in biological systems, it is worthwhile to investigate whether a network of Hopfield neurons is capable of determining correctly the Fourier spectrum of an arbitrary input signal. This requires that the network must be capable of recognizing from a limited set of examples, the various Fourier transform properties concerning linearity, scaling, shifting, and symmetry [13–15]. The shifting and scaling properties, for example, will enable the system to recognize a local signal feature regardless of its position (in space or time) and magnification, respectively.

In the next section, we discuss both the response characteristics of a Hopfield encoding neuron, and the features of the three-layer network architecture being considered. We then investigate by computer experiments if such network is trainable to the task of computing the Fourier spectra of input functions that are part of the training set—the first test of a successful network design [16]. Finally, we carry out the second test which is to find if the trained network can correctly analyze new arbitrary inputs.

II. NETWORK CONFIGURATION

Illustrated in Fig. 1 is the three-layer network with an input layer that consists of $2M$ (integer $M\geqslant 1$) Hopfield encoding neurons that sample a real-valued $s(x)$ of duration T. The Hopfield neurons are equally separated from each other by a distance of $\Delta x = T/2M$, and each neuron is linked to an exclusive decoder. Each decoder is connected to all the $4M$ neurons in the output layer (feedforward architecture [16]).

*Electronic address: csaloma@nip.upd.edu.ph

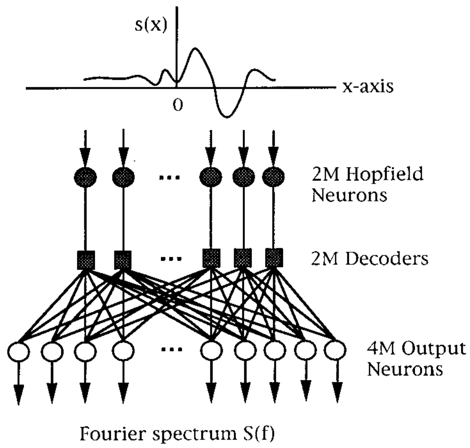

FIG. 1. Schematics of the three-layer, frequency-analyzing network. The $2M$ (integer $M \leqslant 1$) Hopfield encoding neurons which form the input layer are equally separated by a distance $\Delta x = T/2M$. The Hopfield neurons sample the input signal $s(x)$ of length T. Each Hopfield neuron is linked to an exclusive decoder with its output connected to all the $4M$ output neurons. The $4M$ output neurons yield the Fourier spectrum $S(f)$ of $s(x)$. The first $2M$ output neurons yield the real components of $S(f)$ while the next $2M$ outputs give its imaginary components.

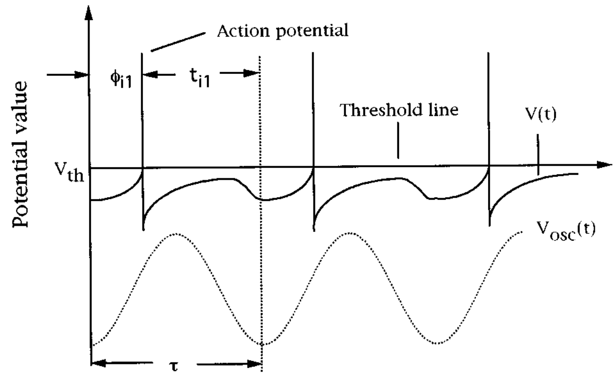

FIG. 2. ith Hopfield neuron generates a series of spikes (also called the action potentials) in response to a specific value $s(i)$ of $s(x)$. The spikes occur at time locations: $\phi_{ik} = k(\tau - t_{ik})$, where $s(i) - V_{\rm osc}(t) = 0$. Integer $k = 1,2,...$ marks successive periods of $V_{\rm osc}(t)$. Only one action potential is generated within each oscillatory period τ.

A neuron in the output layer is hereby referred to as the output neuron.

The first $2M$ output neurons yield the amplitudes of the real components of $S(f)$, while those of the corresponding imaginary components are given by the next $2M$ output neurons. The network, therefore, not only extracts information regarding spectral energy distribution $|S(f)|^2$ of $s(x)$, but also about the relative phases of the component sinusoids of $s(x)$.

Although the $2M$ decoders may be construed to constitute the hidden layer of the network, it must be pointed out that they may not represent actual biological neurons themselves, but rather some synaptic, axonal, or cellular effects which the cell potential $V(t)$ encounters during propagation from its generating Hopfield neuron towards a particular output neuron.

The cell potential $V_i(t)$ of the ith Hopfield neuron ($i = 1,2,\ldots,2M$) is described by [8]

$$V_i(t) - V_{\rm th} = s(i) - V_{\rm osc}(t), \tag{1}$$

where $V_{\rm th}$ is the (fixed-valued) threshold potential, $V_{\rm osc}(t)$ is the time-dependent subthreshold oscillatory drive of the neuron potential, and $s(i)$ represents the particular $s(x)$ value at the input of the ith Hopfield neuron. In our analysis, we assume that the behavior of $V_{\rm osc}(t)$ and the value of $V_{\rm th}$ are the same for all the Hopfield neurons in the network.

The term ''action potential'' refers to the spike (Dirac δ) signal that is generated by the Hopfield neuron when information regarding a particular $s(x)$ value is successfully encoded (see Fig. 2). For simplicity, we use $V_{\rm osc}(t) = B + A[1 - \cos(2\pi t/\tau)]$, where A and B are constants such that no spike is produced when no signal is present. The presence of a nonzero $s(i)$ introduces an upward push (bias) to $V_{\rm osc}(t)$, which, depending on the particular $s(i)$ value, enables it to overcome the threshold barrier set by $V_{\rm th}$. Within a particular period τ of $V_{\rm osc}(t)$, a spike is generated only when $V_i(t) > V_{\rm th}$, and if no other spike has been generated previously within the said period.

For the benefit of a straightforward discussion, we take that the spike is generated exactly (no time lag is involved) at time locations when $s(i) - V_{\rm osc}(t) = 0$. For the ith neuron, the spikes occur at $\phi_{ik} = k(\tau - t_{ik})$, where t_{ik} is called the time advance, and $k = 1,2,...$, etc. For example, ϕ_{i1} denotes the time location of the spike appearing within the first period ($k = 0$) of $V_{\rm osc}(t)$. Every ϕ_{ik} is an algebraic solution to $s(i) - V_{\rm osc}(t) = 0$.

When the $s(i)$ value does not change with time, t_{ik} will be the same for all k values and we write $t_{ik} = t_i (\phi_{ik} = \phi_i)$. In such a case, the spikes generated by the ith Hopfield neuron are equidistant and knowing only the position ϕ_{i1} of the first spike is sufficient to decode the $s(i)$ value. Unless specified otherwise, we deal only with time-independent $s(x)$ signals.

The purpose of the ith decoder is to determine the $s(i)$ value from the spike location ϕ_i. Decoding starts when the spike that has been generated is sensed by the decoder. In the simplest case, the connection weight between a Hopfield neuron and its decoder can be set either to 1 (presence of a transduction pathway from the Hopfield neuron to output neuron), or 0 (no transduction pathway).

The interconnection strength between the ith decoder and the jth output neuron is given by the coupling constant w_{ij} ($j = 1,2,\ldots,4M$). The decoder output $s(i)$ signal that reaches the jth output neuron is attenuated by the value of w_{ij} (positive value for excitation, and negative for inhibitory effects).

All the $2M$ attenuated signals $\{w_{ij}s(i)\}$ arriving at the jth output neuron are summed up within an integration time given by τ because a Hopfield neuron produces only one action potential within each oscillatory period τ. The output potential S_o^j of the jth output neuron is expressed as

$$S_o^j = V_{\rm bias} + \sum_{i=1}^{2M} w_{ij} s(i), \tag{2}$$

where $V_{\rm bias}$ is a potential bias, and $w_{ij}s(i)$ is the attenuated signal received by the jth output neuron from the ith de-

TABLE I. List of all input functions $q(x)$ used in the training of an analyzing network with 32 Hopfield neurons.

Number	$q(x)$	Remarks
32	$\pm\sin(2\pi ux/T)$	$u=1,...,16$
32	$\pm\cos(2\pi vx/T)$	$v=1,...,16$
10	$B\sin(2\pi ux/T)+c\sin(2\pi vx/T)$	Random $B,C<1$; random u,v values
10	$B\sin(2\pi ux/T)+C\cos(2\pi vx/T)$	Random $B,C<1$; random u,v values
10	$B\cos(2\pi ux/T)+C\cos(2\pi vx/T)$	Random $B,C<1$; random u,v values
5	$\exp(-5x)\cos 2\pi(p_1x+p_2x^2)$	p_1 and p_2 are random positive numbers
1	$\mathrm{Tri}(x-0.5)$	Shifted triangular function
1	$f(x)=0$ for $x\leqslant 0.5$ $=\exp\{-5(x-0.5)\}$ for $x>0.5$	Shifted asymmetric function
1	$f(x)=0$ for $x\geqslant 0.5$ $=\exp\{-5(x-0.5)\}$ for $x<0.5$	Shifted asymmetric function
1	$f(x)=0.5$ for $x\geqslant 0.5$ $=\exp\{-5(x-0.5)\}$ for $x<0.5$	Shifted asymmetric function
1	$f(x)=0.5$ for $x\leqslant 0.5$ $=\exp\{-5(x-0.5)\}$ for $x>0.5$	Shifted asymmetric function

coder. We assume that the V_{bias} value is the same for all Hopfield neurons in the network.

Generally, $s(x)$ has no symmetry about $x=0$, and $S(f)$ is complex valued, i.e., $S(f)=S_R(f)+\kappa S_I(f)$, where $\kappa=\sqrt{(-1)}$, $S_R(f)$ and $S_I(f)$ are the real and imaginary part of $S(f)$, respectively. Note that $S_R(-f)=S_R(f)$ and $S_I(-f)=-S_I(f)$ because $s(x)$ is a real-valued function. Because the number of Hopfield neurons in the network is finite, $s(x)$ is only finitely sampled and the computed Fourier spectrum $S_o(f)$ is discretely valued with respect to $f=(-M+1)/T$, $(-M+2)/T,\ldots,M/T$.

For $j=1,2,\ldots,2M$, the outputs $\{S_o^j\}$ represent the real part $\{S_{o-R}(f)\}$ of the Fourier spectrum computed by the network, where $f=(-M+j)/T$. For $j=(2M+1),(2M+2),\ldots,4M$, the outputs $\{S_o^j\}$ represent the computed imaginary Fourier spectrum $\{S_{o-R}(f)\}$, where $f=(-3M+j)/T$.

III. TRAINING

The suitable w_{ij} values were determined by minimizing the error function $e(\mathbf{w})$:

$$e(\mathbf{w})=\frac{1}{2}\sum_{q=1}^{K}\sum_{j=1}^{4M}\{S_o^j(q)-S^j(q)\}^2, \tag{3}$$

which amounted to determining the weight vector $\mathbf{w}_{\min}$ that yields the lowest $e(\mathbf{w})$ value [16]. The components of the weight vector $\mathbf{w}$ are given by $\{w_{ij}\}$. The training set consists of a K number of input-output response pairs. In a given pair, the input consists of $2M$ equally sampled values of the analytic expression $q(x)$ of the qth input function in the training set, while the output consists of the discrete (complex) Fourier spectrum ($2M$ values each for the real and imaginary components) of $q(x)$.

In Eq. (3), $S_o^j(q)$ is the output of the jth output neuron corresponding to $q(x)$, and $S^j(q)$ is the desired (correct) output. The $4M$ elements of $\{S^j(q)\}$ are determined by applying the discrete fast Fourier transform [13] on the $2M$ equally sampled values of $q(x)$. For example, $S^1(q)$ represents the true $S_R(-M+1)$ value corresponding to $q(x)$.

The network was trained using the backpropagation method [16]. Starting from a random set of values, the w_{ij}'s were changed using the steepest descent method [16]: $w_{ij}^{(p)}=w_{ij}^{(p-1)}-\gamma\{\partial e(\mathbf{w})/\partial w_{ij}\}$, where γ is the learning rate, and p is the iteration number.

The time-independent signal $s(x)$, where x represents a position variable, is sampled equally by the $2M$ Hopfield neurons, at a sampling distance of $\Delta x=T/2M$, where T is signal duration. The Fourier spectrum of $s(x)$ can be analyzed at a frequency resolution of $1/T$ and a cutoff frequency $f_c=\pm 0.5/\Delta x=\pm M/T$ [13–15].

Position-independent, temporal signals (i.e., x is a time variable), on the other hand, can be coded with only one Hopfield neuron. In this case, $\Delta x=\tau$ and $f_c=\pm 0.5/\tau$. Thus, component sinusoids of $s(x)$ with frequencies higher than $\pm 0.5/\tau$ are not encoded.

Generally, the locations $\{t_k\}$ of spikes in the $V(t)$ signal of the Hopfield neuron are not equidistant from each other, i.e., $t_1\neq t_2\neq\cdots\neq t_{2M}$. To avoid an undersampling of $s(x)$, the associated decoder must be able to determine the particular $s(x=x_k)$ value from the location t_k of the spike generated within the kth period τ_k $(k=1,2,\ldots,2M)$ of $V_{\mathrm{osc}}(t)$. The decoder performs a total of $2M$ decoding operations within the duration T of $s(x)$.

In our experiments, we analyze time-independent signals ($T=1$) using a network composed of 32 ($M=16$) Hopfield neurons and 64 output neurons with $V_{\mathrm{bias}}=0$. The value of $2M=32$ (instead of 16 or 64) was chosen because it offers the best compromise between frequency resolution and training time for the computer that we were using [IBM PC compatible 486DX-2 (66 MHz)]. The connection strength was set to unity for all the 32 Hopfield neuron-decoder pairs, and the initial values of w_{ij} were randomly selected between -1 and 1 ($\gamma=0.5$).

Listed in Table I are all the input functions $q(x)$ in the training set ($T=1$, $0.5\leqslant x\leqslant 0.5$), which includes all the possible 64 basis functions (32 sine and 32 cosine functions) of

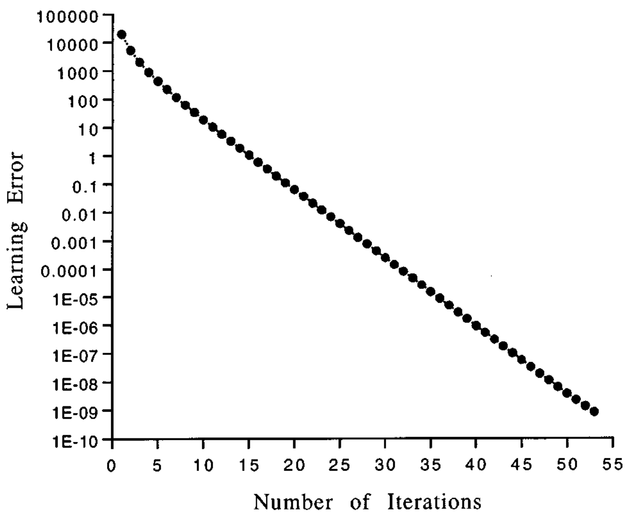

FIG. 3. Exponential decrease of the learning error $e(\mathbf{w})$ with increasing number of iterations ($2M=32$). The same type of exponential behavior was observed for other sets of randomly selected initial weights, or training sets with different arbitrary functions. After 53 iterations, $e(\mathbf{w})$ has decreased to 8.5×10^{-10}.

the possible Fourier series expansion of $s(x)$. The number of general functions was ascertained by noting how the generalization performance of the network improves with the introduction of more arbitrary functions. We utilized the least number of arbitrary functions possible. We observed that for a given M, the choice of the general functions was not unique.

The spike location was determined within each τ, at an accuracy of one in 10^6 (i.e., each τ was divided into 10^6 partitions). The smallest possible change δs of $s(x)$ that can be recognized by the Hopfield neuron is $\delta s=(A2\pi/\tau)\sin(2\pi\tau)\delta t$, where δt is the size of the smallest partition within each τ. The δs value was determined at $t=\tau/2$ where the $V_{\mathrm{osc}}(t)$ is farthest from V_{th}. In our experiments, we used $\delta t=10^{-6}$. The issue of location accuracy is important because all physical systems have a finite time response ($\delta t>0$) and are therefore limited in their ability to detect very weak signals.

Figure 3 shows the behavior of the error function $e(\mathbf{w})$ as a function of the number of iterations, for the 32 Hopfield neuron-network and training set considered. The $e(\mathbf{w})$ curve exponentially decays with increasing iteration number indicating a trainable network [16]. For analyzing signals in the training set (a simple task of memory recall), the trained network computed a Fourier spectrum at an accuracy that depends only on the minimum error value allowed. Equation (2) together with the vector $\mathbf{w}_{\mathrm{min}}$ constitute what we call as the learned solution.

We also investigated the trainability of networks with 16 and 64 Hopfield neurons, respectively, using appropriate sets of training functions and found that their corresponding error functions also decay exponentially with iteration number.

IV. GENERALIZATION

The global validity of the learned solution was tested by letting the trained network analyze other input functions ($T=1$, $-0.5\leqslant x\leqslant0.5$) that were not part of the training set.

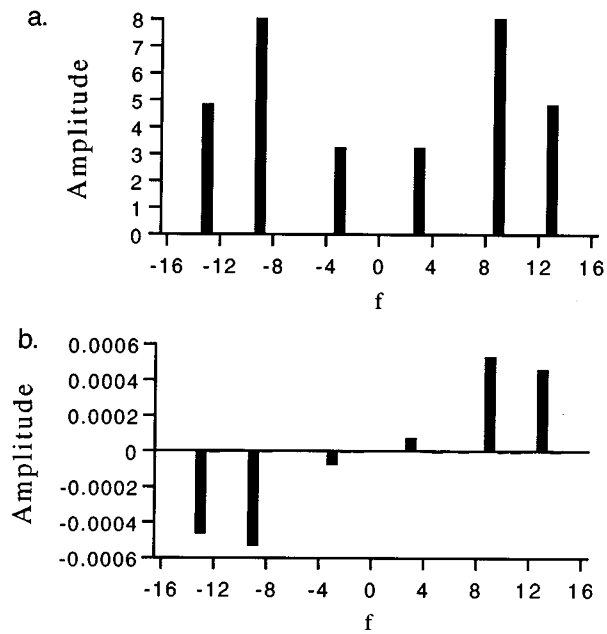

FIG. 4. Network learned the linearity property of Fourier transforms. Computed spectrum corresponding to $s(x)=0.2\cos(2\pi3x)+0.5\cos(2\pi9x)+0.3\cos(2\pi13x)$: (a) $S_{o-R}(f)$ and (b) $S_{o-I}(f)$. The imaginary $S_{o-I}(f)$ values are small because $s(x)$ is an even function ($T=1$).

The generalization performance was measured using the normalized mean-squared error (NMSE): $E_{\mathrm{NMS}}=(\Sigma|S^j-S_o^j|^2)/\Sigma|S^j|^2$, where the summation is carried out from $j=1$ to $j=4M$. The set $\{S^j\}$ represents the true (discrete) complex Fourier spectrum of the new input function where $S^j=S_R(-M+j)$ for $j=1,2,\ldots,2M$; and $S^j=S_I(-3M+j)$ for $j=(2M+1),\ldots,4M$. The set $\{S_o^j\}$ is the corresponding spectrum computed by the trained network.

The results shown in Figs. 4(a) and 4(b) illustrate that the trained network was able to learn the linearity property by correctly computing the discrete Fourier spectrum of $s(x)=0.2\cos(2\pi3x)+0.5\cos(2\pi9x)+0.3\cos(2\pi13x)$, which is not part of the training set. The network recognized the even symmetry of $s(x)$ by yielding negligible imaginary $S_{o-I}(f)$ values where $f=-16,-15,...,16$ ($T=1$). The trained network also performed well with other functional forms including products of two sinusoids. For all the new functions tested, an averaged NMSE value of 2.57×10^{-14} was obtained.

The trained network was also able to learn the scaling and shifting properties by correctly computing the Fourier spectrum of a shifted rectangular function [13,14]: $f_{\mathrm{rect}}(x-0.5)$, and its scaled version $f_{\mathrm{rect}}[k(x-0.5)]$ for $k=0.4$. Figure 5 shows that the computed $[2.5S_o(2.5f)]$ of $f_{\mathrm{rect}}(0.4x-0.2)$ exhibits the correct broadening and relative decrease in the amplitudes of the component sinusoids.

The network detected the 0.5 shift (with respect to $x=0$) in the center position the rectangular function by yielding a complex-valued $S_o(f)=S_{o-R}(f)+jS_{o-I}(f)$. The NMSE values associated with $[2.5S_{o-R}(2.5f)]$ and $[2.5S_{o-I}(2.5f)]$ are 1.04×10^{-14} and 5.83×10^{-14}, respectively.

The trained network also recognized the property that the

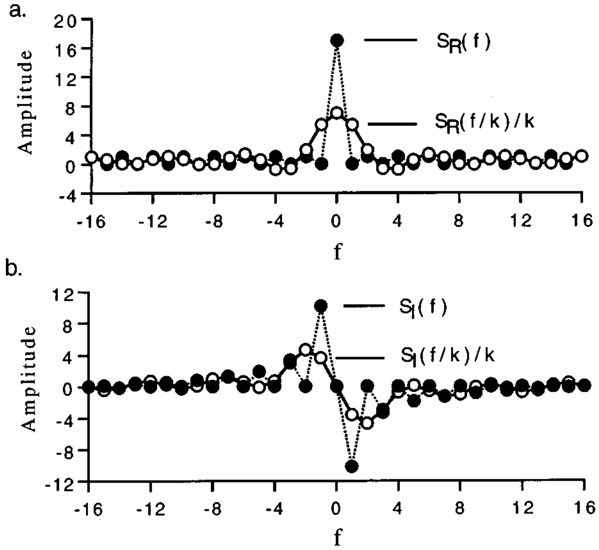

FIG. 5. Network learned the scaling and shifting property of Fourier transforms. Computed $S(f)$ of an input test signal which is a scaled ($k=0.4$) square-wave function that is shifted from $x=0$ by 0.5 ($T=1$): (a) $S_{o-R}(f)$ and (b) $S_{o-I}(f)$. The computed $S_o(f)$'s are complex because both square waves are shifted from $x=0$.

Fourier transform of $S(f)$ yields an inverted image of $s(x)$. Lenses, which are important components in optical imaging systems, yield inverted (real) images because of this property [14].

The image shown in Fig. 6 is a plot of the network output values that were obtained when the computed spectrum (32 discrete values each for S_{o-R} and S_{o-I}): $S_o=S_{o-R}+\kappa S_{o-I}$; of a shifted (original) Gaussian function [13]: $f_{\mathrm{Gauss}}(x-0.5)$, was used as input. The image plot was obtained as follows: (i) Use as network inputs the 32 computed S_{o-R} values and note the values of the 64 network outputs that take the form $S_{o'}=S_{o-R'}+\kappa S_{o-I}$; (ii) Next,

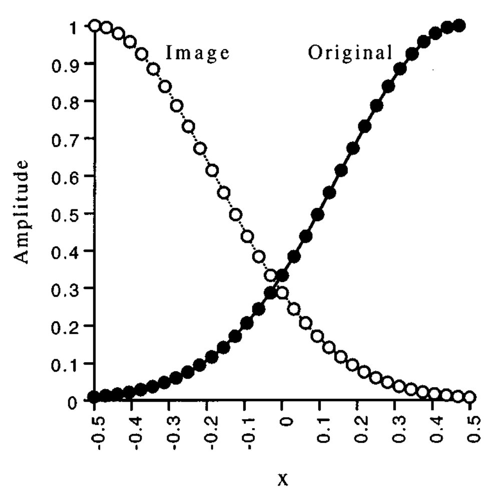

FIG. 6. Network learned that the Fourier transform of $S(f)$ itself is $-s(x)$. The image plot is the network output when the computed $S_o(f)$ of the original shifted Gaussian input signal is utilized as input. See text for details of how the image plot was obtained.

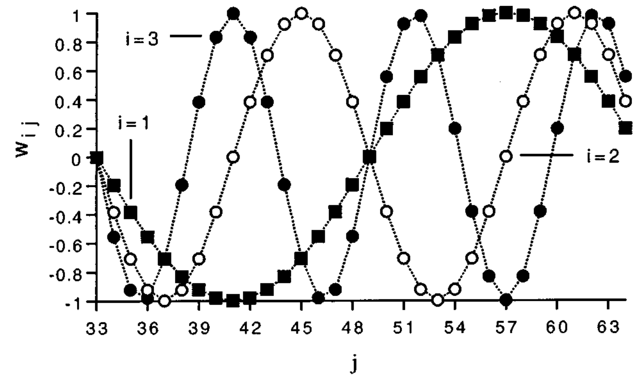

FIG. 7. Plots of the coupling constants w_{ij} vs index i. Indices i and j represent the ith decoder ($i=1,2,...,32$) and jth output neuron ($j=33,34,35$), respectively. The j values are part of the output neurons that yield the imaginary part of $S_o(f)$.

use as inputs the 32 computed S_{o-I} values and note the corresponding output values that take the form $S_{o''}=S_{o-R''}+\kappa S_{o-I''}$; (iii) Calculate the image curve as $(S_{o-R'}+S_{o-R''})+\kappa(S_{o-I'}+S_{o-I''})$.

For $s(x)=f_{\mathrm{Gauss}}(x-0.5)$ as the original input, $[S_{o-I'}+S_{o-I''}]=0$, and the image curve was real ($j=1,\ldots,2M$) and given by the plot of the $(S_{o-R'}+S_{o-R''})$ values. When their asymmetry is neglected, an NMSE of 5.34×10^{-14} was obtained between the original and image curves. The result in Fig. 6 again indicates that the network was able to learn the linearity and the distributive properties of Fourier transformation.

The universal validity of the learned solution can be also seen from the behavior of the final w_{ij} values which form the components of $\mathbf{w}_{\mathrm{min}}$. Figure 7 shows plots of the w_{ij} values that describe the interconnection strengths between the 32 decoders ($i=1,...,32$) and three selected target output neurons ($j=33,34,35$) that yield the imaginary part of the computed Fourier, spectrum.

The w_{ij} curves (for fixed j values), exhibit the behavior of a $\sin(2\pi ij/32)$ function with index i. We have also noted (data not shown) that the w_{ij} plots associated with $j=1,...,32$; exhibit the behavior of a $\cos(2\pi ij/32)$ function. Recall that the first 32 output neurons in the network, yield the real part of $S_o(f)$ and that last 32 output neurons yield the corresponding imaginary part.

Substituting the equivalent sinusoidal expressions of w_{ij} into Eq. (2) yields the terms ($V_{\mathrm{bias}}=0$, summation taken over index i): $S_o^j=\Sigma[s(i)\cos(2\pi ij/32)]$ for $j=1$ to 32, and $S_o^j=\Sigma[s(i)\sin(2\pi ij/32)]$ for $j=33$ to 64. These two derived terms for S_o^j describe the cosine and the sine transforms of $s(x)$ respectively, and it is clear that the learned solution yields the desired network output, which is the (complex) Fourier transform of $s(x)$.

V. DISCUSSION

We have presented a network of Hopfield encoding neurons that is able to recognize the general properties of Fourier transforms from only a limited set of examples. The network was successfully trained using the backpropagation method. No in-depth attempts were made to train the network using evolution-based methods because under such methods training is relatively much slower, and they are therefore used only when backpropagation training fails.

Fourier analysis is a task that is well suited to parallel processing because each Fourier component $S(f)$ is computed using (summation is over i) $S(f) = \Sigma[s(i)\{\cos(2\pi f i \Delta x) - \kappa \sin(2\pi f i \Delta x)\}]$, where $\Delta x = T/2M$ is the sampling interval, $-M \leqslant$ integer $i \leqslant M$, T is the duration of $s(x)$, $f = j/T$, and $-M \leqslant$ integer $j \leqslant M$. Thus, to compute a particular $S(f)$ value one needs to have the $2M$-element set $\{s(i)\}$ which describes the behavior of $s(x)$ over the entire duration T.

Our results also demonstrate that Fourier analysis of $s(x)$ at the same frequency resolution and cutoff frequency, can be achieved by a two-layer, feedforward network consisting of $2M$ input neurons and $4M$ output neurons. In this case, the input neuron performs the combined tasks of coding and then decoding the particular $s(x)$ value at its input. However, no biological analogs of such input neurons have been observed so far although it is interesting to note that information coding by nonspiking neurons have recently been found in the visual system of blowflies [17]. The nonspiking neuron yields an output (often distorted) that directly describes the amplitude behavior of the input signal.

The encoding scheme employed by a Hopfield neuron is notably similar to the manner in which sinusoid crossing (SC)-based spectral analysis is implemented [18,19]. This is particularly apparent when $s(x)$ is a time-based signal where the coding of $s(x)$ can be achieved with only one Hopfield neuron.

In SC-based analysis, $S(f)$ is computed directly from locations $\{x_i\}$, where $s(x)$ intesects with a reference sinusoid $r(x) = A\cos(2\pi M x/T)$. The SC's in $\{x_i\}$ are labeled from $i=1$ to $2M$ according to their order of detection relative to $x=0$. Each SC location x_i is an algebraic solution to $s(x) - r(x) = 0$. If $A \geqslant |s(x)|$ for all possible $s(x)$ values within T, then there will be $2M$ crossings with one SC occurring within each interval $\Delta x = T/2M$ of $r(t)$.

The information content of $s(x)$ is completely encoded in $\{x_i\}$, and it has been shown that in addition to $S(f)$, the Hartley [20] and wavelet transform [3] of $s(x)$ can also be computed directly from $\{x_i\}$. An advantage of SC sampling over the amplitude sampling of $s(x)$ at equal intervals of x is in the simplicity of the required hardware support. Only a single comparator is needed in an SC detector [21] in contrast to several in a conventional analog-to-digital converter.

For the Hopfield neuron, $V_{\text{osc}}(t)$ takes the role of $r(x)$, and each location ϕ_k of the action potential peaks satisfies the condition $s(x) - V_{\text{osc}}(t) = 0$, provided $V_{\text{th}} = 0$. Note that $\{\phi_k\}$ and $\{x_i\}$ are identical if x is a time variable, which suggests that temporal coding using $\{\phi_k\}$ is a versatile scheme of representing the information content of an external stimulus in a neural network. Note, however, that in SC-based spectrum analysis, the $S(f)$ components are computed iteratively, unlike that in the analyzer network.

[1] B. Mel, Neural Comput. **6**, 1031 (1994).

[2] A. Longtin and K. Hinzer, Neural Comput. **8**, 215 (1996).

[3] C. Saloma, Phys. Rev. E **53**, 1964 (1996).

[4] W. Singer, Science **270**, 758 (1995).

[5] E. Adrian, *The Basis of Sensation* (Christophers, London, 1928).

[6] J. Maunsell, Science **270**, 764 (1995).

[7] L. Ungerleider, Science **270**, 769 (1995).

[8] J. J Hopfield, Nature **376**, 33 (1995).

[9] T. Bullock, Rev. Neurosci. **16**, 1 (1993).

[10] N. Burgess, J. O'Keefe, and M. Recce, in *Advances in Neural Information Processing Systems*, edited by S. Hanson, C. Giles, and J. Cowan (Morgan Kaufman, San Mateo, CA, 1993), Vol. 5, pp. 929–936.

[11] F. Heiligenberg, *Neural Nets in Electric Fish* (MIT, Cambridge, MA, 1991).

[12] C. Carr and K. Konishi, J. Neurosci. **10**, 3227 (1990).

[13] R. Bracewell, *The Fourier Transform and Its Applications* (Prentice-Hall, Englewood Cliffs, NJ, 1988).

[14] J. Goodman, *Introduction to Fourier Optics* (Wiley, New York, 1968).

[15] D. Manolakis and J. Proakis, *Digital Signal Processing: Principles, Algorithms, and Applications* (Macmillan, New York, 1992).

[16] T. Watkin and A. Rau, Rev. Mod. Phys. **65**, 499 (1993).

[17] J. Haag and A. Borst, Nature **379**, 639 (1996).

[18] C. Saloma and P. Haeberli, Opt. Lett. **16**, 1535 (1991).

[19] C. Saloma, J. Appl. Phys. **74**, 5314 (1993).

[20] C. Saloma, Opt. Lett. **20**, 1 (1995).

[21] C. Saloma and V. Daria, Opt. Lett. **18**, 1468 (1993).