

# Detection of Pitch in Random Acoustic Signals by Neural Networks

**Nicola Cufaro Petroni\***

**Franco Degrassi<sup>†</sup>**

and

**Guido Pasquariello<sup>‡</sup>**

By means of a backpropagation neural network a model has been built which is able to distinguish between noises and tones endowed with a detectable pitch in a (computer simulated) random acoustic environment where the information carried by the signals is compressed to its essential part by the reduction of the Fourier transform into templates of 12 numbers used as inputs. It is found that a neural network able to detect a pitch is also able to recognize the presence of a residue pitch in the signals of complex tones where the first (or the first two) harmonic has been subtracted. Finally the correlations between the concept of consonance and the presence of a detectable pitch in the superposition of pairs of complex tones are briefly investigated.

Revised version: Bari, June 1994

To be published in *Journal of New Music Research*

---

\* Dipartimento di Fisica dell' Università di Bari and G.N.C.B. Unità di Bari; via Amendola 173, 70126 BARI (Italy); E-mail CUFARO@BARI.INFN.IT.

<sup>†</sup> G.N.C.B. Unità di Bari; via Amendola 173, 70126 BARI (Italy).

<sup>‡</sup> I.E.S.I.- C.N.R.; via Amendola 166/5, 70126 BARI (Italy).

## 1. INTRODUCTION

The ultimate aim of this research is to try to account for the emergence of the tonal sensitivity in the tighter connection possible with the physical phenomena of the sound. However we must immediately acknowledge that the results accounted for in this paper describe only a very preliminary stage of this investigation and that, for the time being, we are not trying to build a psychological model for tonal relations: the aim of this paper is much more limited in scope and it is rather that of building a model to test the ability of a *neural network* in extracting some particular information from a particular *random environment*. This will amount also to a test of what sort of information is really contained in the random environment simulated, so that this will also be a test to verify the hypotheses needed to build it. It is clear, from the previous remarks, that in this paper the emphasis will be on *physics* and *information* rather than on *psychology* and *music*, even if, in our opinion, this will shed some light also on the birth of a tonal sensitivity.

In trying to approach an understanding of musical ideas about the tonal system there are different levels of investigation: pitch, consonance, keys, tonal relations ... and so on. For the time being we will limit ourselves only to the first part of this sequence and we will try to investigate *how much in the higher levels* is explainable in terms of concepts of the *lower levels*. More precisely we will try to understand what conclusions it is possible to draw from the ability of individuating the presence of a *pitch* in an acoustic signal when this ability is considered as a primary fact. In practice the idea is to embed an *adaptive subject* (in our case a backpropagation neural network) in a *random acoustic environment*. The training will consist in the fact that the network will learn to discriminate between signals with or without a pitch. Of course the definition of the random acoustic environment will be crucial to establish the ability to generalize of the neural network, and, on the other hand, it should also be an acceptable model of what we can find in a *natural* acoustic environment.

The idea of pitch is a psychological one meaning the perceived *height* of an acoustic signal: in a sense the perceptive difference between a *noise* and a *tone* is that in the former you can not distinguish a pitch, while in the last you can. Of course there is a continuum of possibilities between these two extreme situations and a lot of ambiguous signals; however the real world is full of signals perceived with or without a clear pitch. In fact, among the signals coming from the natural world there are some produced by specific physical systems with particular boundary conditions (strings, plates, bells, tubes, voices,...) which at a different degree can be described as a superposition of *harmonic* or *quasi-harmonic* partials. This can, in a sense, be referred to as the fact that these signals contain a particular information: *the reference to a pitch or fundamental frequency* (albeit not to the *lower* frequency actually present in the signal). Others signals (coming

for example from wind, falling water, rustles,...) have no such characteristic and a *height* of the signal can be perceived only with the outmost difficulty, if at all.

Can this particular information, encoded in the complex signals of the real world, be detected by means of a simple model based on a neural network? In which *part* of the signal is the essential of this information encoded? How much this ability in detecting a pitch can be generalized to more complex situations? These are the questions that we are trying to answer in this paper, with the idea that the ability in detecting the presence of a pitch can say something also on our tonal sensitivity. Even if it is clear to everyone that the tonal system is not based (or at least *not only based*) on the presence of a pitch, we can nevertheless say that the reference to a *tonic* is one of the building blocks of the classical tonality. In this paper we do not bring the discussion to the level of chords or sequences of chords where much more complex problems are involved; but our opinion is that, at a very elementary and physical level, one of the starting points in the building of the tonal system is the ability of individuating a pitch in the acoustic signals.

It is well known (Zwicker E. and Fastl H. 1990) that frequency alone is not sufficient to describe the pitch produced by pure tones: it depends, for example, on their intensity. However, even for complex tones, the sensation of pitch can be considered as essentially connected to the nature of their spectrum, namely to the Fourier transform of the signal (of course we suppose, for the time being, that our tones are steadily produced for a time long enough to have a determination of the pitch: at present this is just a *static* study) and in particular to its modulus. By stressing the relevance of the amplitude of the Fourier transform of the signal in some sense we make the implicit hypothesis that the principal pitch perception mechanism is the so called *place theory* (Hall D. H. 1982) but the following discussion will show that our idea is better described as a mixing of the place theory and the more recent *pattern recognition theories*. In fact all the information connected to the pitch (and to all the other characteristics of the signal) is encoded in the Fourier transform, but this must be considered a highly redundant information if we are interested uniquely to the pitch identification. Hence we will introduce a few *hypotheses* in order to eliminate this redundancy by reducing the complete Fourier transform to the features that we consider essential to the pitch identification. Of course this procedure of reduction will also introduce ambiguities in the sense that, for example, the same reduced Fourier transform can be produced starting from very different (harmonic and non-harmonic) initial Fourier transforms: every operation of *projection* like this one will introduce similar ambiguities, in the sense that in general we can not uniquely retrieve the initial object starting from a projected image of it. However this work will also be considered as a test of this choice for the representation of the acoustic signals: the reduction of the Fourier transform will indeed be based on some hypotheses on what we think to be (in a first approximation) unessential to the detection of a pitch and the results of this work can hence also be seen as a

verification of these hypotheses. In other words, at this preliminary stage of the research, our aim is less the excellence of the performance of the network in distinguishing between noises and tones than to test how much information about the pitch is lost in the compression of the signal to a few numbers (the reduced Fourier transform); otherwise we would have designed our system in a very different way. On the other hand we think that this preliminary inquire is important since, in later stages, it will be relevant (also to save CPU time) to know what in a real signal can be considered *redundant* for the pitch extraction, and hence what compression operations are allowed with a negligible loss of information.

In order to build our representation, we will first of all *discretize* the *continuous* set of all our frequencies. That will be done not only in order to realize computer simulations, but also in order to individuate the *discrete* elements of our language: the tones of a diatonic scale. In fact the discretization will be realized by means of a series of tones all at the same *well tempered distance* of  $100\phi$ . The modulus of the Fourier transforms will be reduced to templates of weights attributed only to the frequencies of this *well tempered* series of tones.

Secondly we will suppose that *the phases of the different partial vibrations are not essential* for our aims, so that the number assigned to every diatonic tone of our series will be a positive real number corresponding to the modulus of a (complex) Fourier transform.

Finally we will suppose a complete *octave equivalence* by means of what we will be able to reduce all our Fourier transform in just one octave interval  $[\nu, 2\nu)$ , the exact location of  $\nu$  being not relevant for our purposes. In this way we will achieve a sort of reduction of all the frequencies to only 12 *pitch classes* contained in the chosen octave.

Hence in this study a signal (harmonic or noisy) will be represented by means of this *reduced Fourier transform*, namely by means of the template of 12 numbers described above and the pitch detection will be considered as a problem of *pattern recognition* on these templates. This point is important since one of the aims of this research will be to investigate if a neural network, trained to discriminate signals with a pitch from signals without a pitch, can also attribute a pitch to tones lacking a few partials which are not included in the training set of examples. Finally a second generalization to be investigated will be that of the trained net to perceive superpositions of *consonant* or *dissonant* tones as respectively reinforcing or disturbing the pitch detection.

## 2. DESCRIPTION OF THE MODEL

Our model will be composed of an *adaptive listener* embedded in a *random acoustic environment* supplying external stimuli. This section will be devoted to the description of these two constituting elements.

**2.1 The Adaptive Listener:** A very simple model for a real listener will be simulated by means of a *backpropagation neural network* (Hecht-Nielsen R. 1989) which will undergo a supervised training in the random acoustic environment. A backpropagation neural network is a computing system composed of *distributed* and *parallel* processing elements called *neurons* (by analogy with the natural neurons in a biological nervous system) *distributed* in *layers* which are fully connected by so called *synapses*. If we consider, for example, the  $j$ -th neuron, the synapses arriving on it will constitute a set  $w_{ij}$  of real numbers, where the index  $i$  will vary on the set of indexes of the neurons of the previous layer. These neurons of the previous layer will transmit to the  $j$ -th neuron their outputs  $x_i$  weighed by means of the synaptic values, and the  $j$ -th neuron will compute a weighed sum of these inputs:  $z_j = \sum_i x_i w_{ij}$ . Then  $z_j$  will become the argument of a function  $y_j = f_j(z_j)$  which will be the output signal of the  $j$ -th neuron and which will be fanned out to all the neurons of next layer weighed by means of the relative synapses. Hence, if we give some input values to the first layer of neurons, the activation of the processing elements will propagate itself from layer to layer until to the final layer where the output will be collected.

In the following we will use backpropagation networks whose neuron activations can take any real value in the interval  $[0, 1]$ . The behaviour of the net as a computing system is determined by the matrix  $w_{ij}$  of the synapses and by the form of the functions  $f_j$ : if these elements are fixed, the net implements a particular function in the sense that, if the input layer is constituted of  $n$  neurons and the output layer of  $m$ , for every input vector  $x = \{x_i\}_{i=1,\dots,n}$  we obtain an unique output vector  $y' = \{y'_j\}_{j=1,\dots,m}$ ; namely we can say that the net implements the function  $F(\cdot; w) : R^n \rightarrow R^m$ .

However the decisive feature of a backpropagation neural network is that it is an *adaptive* computing device. This means that there are algorithms allowing a progressive modification of the synapses and of the threshold values of the network until a particular performance of the outputs is achieved. This is more often implemented by means of a *supervised training*: the outputs  $y'^{(k)}$  ( $k = 1, \dots, N$ ) of the network corresponding to a set of  $N$  input data  $x^{(k)}$  ( $k = 1, \dots, N$ ) are compared with a set of *required* outputs  $y^{(k)}$  supplied from the outside and the network parameters are progressively modified so that the outputs will come nearer and nearer to the required outputs. Of course here two questions arise: a) what *near* means for our vectors  $y$ , and b) how to build an algorithm *convergent* toward the required values.

The usual answer to the first question is: *near* means near *in mean square*. In fact the problem in this form is correctly described as a problem of *approximation* (Poggio T. and Girosi F. 1989): we have some information about the function  $f$  coming from a set of  $N$  *examples*, namely  $N$  pairs  $\{x^{(k)}, y^{(k)}\}_{k=1,\dots,N}$  and we must *reconstruct* the function on the ground of these data. At this

effect we require that the following distance (mean square deviation)

$$\frac{1}{N} \sum_{k=1}^N |F(x^{(k)}; w) - y^{(k)}|^2 \quad (2.1)$$

be minimal.

As for the second question the usual answer is the so called algorithm of the *backpropagation of errors* (Hecht-Nielsen R. 1989; Rumelhart D. E. and McClelland J. L. 1986). In short this means that the optimization of the result is achieved by means of a sequence of iterations: at the end of every cycle the outputs  $y^{(k)}$  of the network are compared with the required outputs  $y^{(k)}$  and the errors  $|y^{(k)} - y^{(k)}|^2$  are taken into account by means of some particular rule (the Delta rule, the Steepest descent rule, ...) in order to modify the synaptic values  $w$  so that in the next iteration a smaller error is realized. Details about these algorithms can be found in the literature: all we need to say here is that they guarantee to achieve a good (if not perfect) approximation in mean square of the function  $f$  which is to be approximated.

However we must remark that in this paper we consider the backpropagation neural networks from a standpoint which is different from the usual one. In fact we will suppose that  $X$  and  $Y$  are *random vectors* representing respectively a random signal (or better its reduced Fourier transform) coming from the outer world and the required answer which is also random. However here something more must be said about what exactly our neural network is now required to do.

Usually a neural network is supposed to approximate a given function; but, even when between the random vectors  $X$  and  $Y$  there is a *statistical dependence*, this does not mean at all a *functional dependence*. We will say that  $X$  and  $Y$  are functionally dependent when there is some function  $f: R^n \rightarrow R^m$  such that  $Y = f(X)$ . This requires that just one value  $Y$  be associated to every value  $X$ ; but in general this is not the way in which statistically dependent random vectors behave, since it can happen that to every value of  $X$  many values of  $Y$  are associated following some distribution law\*. Hence, in the general case of statistical dependence, our problem can not be *to find the functional relation between  $X$  and  $Y$* , since this simply does not exist; rather the right position of the problem will be *to estimate  $Y$  by means of  $X$* , namely to find another random vector  $Y'$  which is both a function of  $X$  and the best approximation of  $Y$ . It is well known (Shiryayev A. N. 1984) that the general answer to this problem is (for every component of our random vectors)

$$Y'_i = E(Y_i | X) = g_i(X); \quad i = 1, \dots, n$$

---

\* Think, for example, to the case of the random variables  $H$ , altitude in the atmosphere, and  $T$ , air temperature. It is well known that there is a dependence of  $T$  on  $H$ : the more we go up, the more the temperature goes down. However it is also clear that this dependence is only statistical and not functional: at a given value of  $H$  does not correspond a unique value of  $T$  since the temperature depends on many other variables that we are neglecting here (latitude, atmospheric conditions, hour of the day,...), so that no function  $f$  such that  $T = f(H)$  can exist.

where  $g_i(x) = E(Y_i | X = x)^*$ . However it is also well known that to find the functions  $g_i(x) = E(Y_i | X = x)$  is not at all an easy task and hence it is remarkable that a backpropagation neural network, since it works exactly on the principle of the mean square approximation, can be considered as computing system which implements a good approximation of the function  $g_i(x) = E(Y_i | X = x)$ . Of course in this case we must think to our training examples as the values  $(x^{(k)}, y^{(k)})_{k=1, \dots, N}$  of a random sample  $(X^{(k)}, Y^{(k)})_{k=1, \dots, N}$  obtained by means of  $N$  repeated (and independent) measurements of the couple of random vectors  $(X, Y)$ , and to our training procedure as a method to find the value of  $w$  which minimizes the value of (2.1). However in this case, differently from that of the simple approximation of functions, we must also take into account situations in which we have  $x^{(k)} = x^{(l)}$  and  $y^{(k)} \neq y^{(l)}$ , namely: to the same input data we can associate several different outputs. The relevance of these remarks for our work is stressed by the fact that we use as inputs sets of templates obtained as reduced Fourier transforms of a signal, and hence we must take into account the case in which identical (or almost identical) templates, produced by very different signals, require totally different answers. This, of course, will be much less important if we used the complete signals (or spectra), but what we want to test is exactly *how much* the inevitable loss of information produced in the projection of the signal into its reduced Fourier transform is relevant with respect to the separation of the *classes* of noises and tones with a pitch and hence the fact that the identification of these classes is now *fuzzy* is an unavoidable feature of the model.

In other words the signal  $x(t)$  is just one possible trajectory of a stochastic process  $\xi(t)$ . We will suppose first of all that there is some functional  $F[\cdot]$  defined on the space of the trajectories  $x(t)$  and taking values in  $[0, 1]$  which represents a measure of the presence of a detectable pitch, so that  $Y = F[\xi(\cdot)] \in [0, 1]$  will be the random variable which we are interested in when we look for something indicating the presence of a pitch. However on the one hand we do not know the form of  $F[\cdot]$  and on the other the complete signal (or its complete Fourier transform) is somehow redundant with respect to the information needed to detect the presence of a pitch. Hence by means of a suitable transformation  $T[\cdot]$  we will reduce our initial signal to somewhat much more simple: the reduced Fourier transform, namely the random vector  $X = \{X_l\}_{l=0, \dots, 11} = T[\xi(\cdot)] \in [0, 1]^{12}$ . Of course now we can not hope to find still another *functional* relation<sup>†</sup> between  $X$  and  $Y$ , but we can at least think that (if our model is meaningful) they will be correlated so that we can *estimate*  $Y$  through  $X$ . The best (in mean square) estimation is  $Y' = E(Y|X)$  and such conditional expectation is just what is supposed to be implemented by our neural network.

---

\* In the following the symbols  $E(\cdot)$ ,  $V(\cdot)$  and  $E(\cdot|\cdot)$  will represent respectively expectation values, variances and conditional expectation values of random variables.

† In fact, by eliminating the redundancy of the signal, we have introduced some ambiguity in the sense that, as already remarked, the same reduced Fourier transform can be obtained from different signals associated to different values of  $Y$ .

**2.2 The random acoustic environment:** In order to build our random acoustic environment we will first of all discretize the frequency axis. The fundamental objects representing our acoustic signals will be a complex function of the frequency  $\nu$ , namely the Fourier transform  $\phi(\nu)$  of our signal  $x(t)$  considered as a function of the time  $t$ . We will limit ourselves only to a sequence of discrete values of the frequency, namely

$$\nu_k = 2^{k/12} \nu_0; \quad k = 0, \pm 1, \pm 2, \dots$$

which are  $100 \phi$  (namely a *well tempered semitone*) apart so that they constitute a sort of (extended) well tempered keyboard. If our signal has a *continuous spectrum*, namely if its Fourier transform is a continuous function of  $\nu$ , we will approximate it by means of a discrete (in general non harmonic) spectrum by assigning a weight  $\phi_k$  only to the points  $\nu_k$ . We then introduce our hypothesis about the *octave equivalence*: we will assimilate all the frequencies  $\nu_k$  which are one or more octaves apart, and we will take just one of these frequencies as a representative of the corresponding *pitch class* and finally we will attribute to this frequency the combined weight of all the elements of this class. As representatives of the 12 pitch classes we will choose the frequencies  $\nu_l = 2^{l/12} \nu_0$ , ( $l = 0, 1, \dots, 11$ ) and we will attribute to every one of these the weights of the elements of the same pitch class. For every  $\nu_k$  it is easy to see that the corresponding pitch class representative  $\nu_{l_k}$  (with  $l_k = 0, 1, \dots, 11$ ) is defined by the relation  $l_k = k \bmod 12$ , where  $n \bmod m$  indicates the (positive) remainder of the division  $\frac{n}{m}$ , while to a given  $l \in \{0, 1, \dots, 11\}$  will correspond the following sequence of values of  $k$ :  $k_l(j) = l + 12j$ ; ( $j = 0, \pm 1, \pm 2, \dots$ ). Hence to every  $\nu_l$ ; ( $l = 0, 1, \dots, 11$ ) we will associate the weight (namely the share of the signal energy which is attributed to this pitch class)

$$x_l = \frac{1}{M} \sqrt{\sum_{j=-\infty}^{\infty} \phi_{l+12j}^2}, \quad M^2 = \sum_{k=-\infty}^{\infty} \phi_k^2, \quad (2.2)$$

and the 12 component vector  $x = \{x_l\}_{l=0, \dots, 11}$  will be considered as our *reduced Fourier transform*. Let us remark that with this definition the 12 numbers of every template representing a signal are *normalized*, namely that  $\sum_{l=0}^{11} |x_l|^2 = 1$ . This means in practice that we consider signals with the same total energy in order to eliminate every dependence on the loudness (intensity) of the signal. In order to avoid formal complications we will also suppose that all our signals are limited in band (a fair supposition because of the human audibility limits) so that we will not consider all the infinite sequence of the  $\nu_k$ : more precisely, if for example  $\nu_0 \simeq 622.3Hz$  (which corresponds to the E<sub>5</sub>b) and we take  $k = -60, -59, \dots, 0, 1, \dots, 59$ , we will have 120 well tempered frequencies running from about 20 Hz through about 20 KHz. With this limitation the series in (2.2) becomes a finite sum so that no problem of convergence will arise.

When our signal is periodic (along an infinite interval of time) the Fourier transform reduces itself to a Fourier series and is described by means of a *discrete harmonic* (or quasi-harmonic, if the

signal is produced through a fairly limited interval of time) *spectrum*. If  $\nu^{(1)}$  is the fundamental frequency of this spectrum, the sequence of the harmonic partials is  $\nu^{(n)} = n\nu^{(1)}$ ;  $n \geq 1$ , and it is clear that in general these frequencies will not exactly coincide with one of our well tempered  $\nu_k$ . Even if  $\nu^{(1)}$  coincides with one among the  $\nu_k$ , the remaining partials will not in general behave in the same simple way. Hence, even in the case of periodic signals, we must somehow approximate our spectrum in order to define the right template of weights for the 12 pitch classes. Indeed we will attribute the modulus of the amplitude of the  $n - th$  partial component to the nearest (in  $\phi$ ) well tempered frequency  $\nu_{k_n}$ . It is very easy to verify that  $k_n - k_1$  turns out to coincide with the integer number nearest to  $12 \log_2 n$ . Finally, by means of the octave equivalence, we will project the frequencies  $\nu_{k_n}$  on the correspondent pitch classes by means of the rule  $l_n = k_n \bmod 12$ , and we will attribute to every pitch class the usual superposition of the weights of the corresponding partials.

For the sake of simplicity we will consider only harmonic signals whose fundamental frequency is one of the well tempered frequencies (namely  $\nu^{(1)}$  coincides with one of the  $\nu_k$ ) and we will characterize the pitch class  $\nu_{l_n}$  of every partial  $\nu^{(n)}$  in musical notation with its location along a diatonic scale relative to the fundamental pitch class  $\nu_{l_1}$ . In other words we will have:

<i>relative location</i>	<i>pitch class</i>
<i>I</i>	fundamental
<i>I</i> ♯	1 semitone above the fundamental
<i>II</i>	2 semitones above the fundamental
<i>III</i> ♭	3 semitones above the fundamental
<i>III</i>	4 semitones above the fundamental
<i>IV</i>	5 semitones above the fundamental
<i>IV</i> ♯	6 semitones above the fundamental
<i>V</i>	7 semitones above the fundamental
<i>VI</i> ♭	8 semitones above the fundamental
<i>VI</i>	9 semitones above the fundamental
<i>VII</i> ♭	10 semitones above the fundamental
<i>VII</i>	11 semitones above the fundamental

**Tab. 2.1**

With this notation the attribution of pitch classes to partials is the following:

$n$	$12 \log_2 n$	$k_n - k_1$	<i>relative location</i>
1	0.00	0	<i>I</i>
2	12.00	12	<i>I</i>
3	19.02	19	<i>V</i>
4	24.00	24	<i>I</i>
5	27.86	28	<i>III</i>
6	31.02	31	<i>V</i>
7	33.69	34	<i>VII</i> ♭
8	36.00	36	<i>I</i>
9	38.04	38	<i>II</i>
10	39.86	40	<i>III</i>
11	41.51	42	<i>IV</i> ♯
12	43.02	43	<i>V</i>
13	44.40	44	<i>VI</i> ♭
14	45.69	46	<i>VII</i> ♭
15	46.88	47	<i>VII</i>
16	48.00	48	<i>I</i>
17	49.05	49	<i>I</i> ♯
18	50.04	50	<i>II</i>
19	50.97	51	<i>III</i> ♭
20	51.86	52	<i>III</i>
21	52.71	53	<i>IV</i>
22	53.51	54	<i>IV</i> ♯
23	54.28	54	<i>IV</i> ♯
24	55.02	55	<i>V</i>
25	55.73	56	<i>VI</i> ♭
26	56.40	56	<i>VI</i> ♭
27	57.06	57	<i>VI</i>
28	57.69	58	<i>VII</i> ♭
29	58.29	58	<i>VII</i> ♭
30	58.88	59	<i>VII</i>
31	59.45	59	<i>VII</i>

**Tab. 2.2**

As can be seen from the previous table, we will consider the partials of a periodic signal contained in the first 5 octaves in order to be sure that no pitch class will have exactly zero amplitude: some exactly zero amplitude (a case never found in practice) could become a too evident feature of the harmonic spectra so that the simulation could be inherently biased. Remark also that the proposed attribution can be found also in every book of musical acoustics (Hall D. E. 1982). Finally this is the list of the partials attributed to every location in the diatonic scale with respect to the fundamental frequency:

<i>relative location</i>	<i>attributed partials</i>
<i>I</i>	1, 2, 4, 8, 16
<i>I</i> ♯	17
<i>II</i>	9, 18
<i>III</i> ♭	19
<i>III</i>	5, 10, 20
<i>IV</i>	21
<i>IV</i> ♯	11, 22, 23
<i>V</i>	3, 6, 12, 24
<i>VI</i> ♭	13, 25, 26
<i>VI</i>	27
<i>VII</i> ♭	7, 14, 28, 29
<i>VII</i>	15, 30, 31

**Tab. 2.3**

Following this table it is now possible to attribute a weight  $x_l$  to every pitch class. For example, if  $\nu_{l_1}$  (with  $l_1 \in \{0, 1, \dots, 11\}$ ) is the pitch class of the first partial of our periodic signal, and if  $\{\phi_n\}_{n=1, \dots, 31}$  are the amplitudes of the partials contained in the first 5 octaves, we will attribute to the pitch class  $\nu_{l_1}$  the weight

$$x_{l_1} = \frac{1}{M} \sqrt{\phi_1^2 + \phi_2^2 + \phi_4^2 + \phi_8^2 + \phi_{16}^2},$$

where, as in the previous case,  $M$  represents the total energy of the signal defined as

$$M^2 = \sum_{n=1}^{31} \phi_n^2.$$

The other attributions of weights will be done in an analogous way, and hence the reduced Fourier transform will be defined even in the case of periodic signals.

### 3. SIMULATION

In this section we will discuss in some detail how our model has been simulated on a computer. In particular we will spend some time in the description of the production of the samples, namely of the random signals that we need in order to train and to test the performances of the neural network. Since this is connected to the simulation of a realistic acoustic environment, it will be very important to design a fair way of producing our samples.

**3.1 Production of the samples:** Our first problem will be that of the way in which we will pick up at random a periodic signal. It is well known that every periodic signal  $x(t)$ , under not very restrictive analytical conditions, can be represented by means of a trigonometric Fourier series (Courant R. and Hilbert D. 1953, Vol. I)

$$x(t) = \sum_{n=0}^{\infty} r_n \cos(2\pi n\nu t + \theta_n).$$

If now we want to pick up at random a periodic signal we can consider the stochastic process  $\xi(t)$  given by the *random trigonometric series*

$$\xi(t) = \sum_{n=0}^{\infty} \xi_n \cos(2\pi nvt + \zeta_n)$$

by substituting respectively the numbers  $r_n, \theta_n$  with the sequence of independent random variables  $\xi_n, \zeta_n$  ( $\xi_n \geq 0; 0 \leq \zeta_n < 2\pi; \forall n \in \mathbb{N}$ ). It must be remarked here that, since in all our simulations we will produce only a *finite* number of terms of every Fourier series, all our series will be finite and hence will trivially *converge*. However we want to stress that a *realistic* signal can be approximately represented by this model only when our finite sums are the first part (the first 31 terms) of a *convergent* series. Hence it is important to know how the terms of convergent series behave if we want to simulate the periodic signals in an acceptable way. For example, if  $\xi$  is a random variable uniformly distributed in  $[0, 1]$ , and we define the following sequence of independent random variables

$$\xi_n = \frac{1}{n^p} \xi; \quad (p > 1; K > 0; n \geq 1) \quad (3.1)$$

it can be shown that our random trigonometric series will be everywhere convergent. Even if this is not the more general way to get random periodic signals, we can nevertheless suppose that the spanned set of functions will be fairly large to contain at least good approximations of the real signals. Let us also remark that for  $p \leq 1$  we can say nothing about the convergence of the random trigonometric series. However, since the aim of the present research is also to explore the ambiguous region between harmonic tones and noises, in order to produce tones with an unusual blend of harmonic partials (*noisy tones*) we will introduce in the following of this paper also values of  $p \leq 1$ . In this region (where the reduced Fourier transform of a noisy tone is not very different from that of a noise with a narrow bandpass) the judgement on the pitch is much more difficult and hence the problems of pattern recognition are much more interesting in the sense that the test of our hypotheses on the conservation of the essential information in the compression of the signal will be more severe.

Basically we have to produce two different sort of *signals*: that with a detectable pitch coming from periodic functions of time and that without a detectable pitch coming from non periodic functions of time. The random signals with detectable pitch will be simulated by means of random trigonometric series. The particular pitch class of the signal will also be random; indeed we will take at random an integer number  $k \in \{-60, \dots, 59\}$  and we will consider the corresponding Fourier series with fundamental frequency  $\nu_k$ . Of course we will consider only the amplitudes of the first 31 partials which fall in the audibility interval  $[\nu_{-60}, \nu_{59}]$ : to determine these numbers we will first of all produce 31 random real numbers  $\{\phi_n\}_{n=1, \dots, 31}$  uniformly distributed in  $[0, 1]$ ; then in order

to have the amplitudes we have several possible choices: we can produce *noisy* tones with a pitch (namely periodic signals with a strong presence of higher partials) simply by taking  $c_n = \phi_n$  as our amplitudes without imposing any modulation for the higher frequencies (namely we choose  $p = 0$  in the random variables  $\xi_n$ ). Usual periodic signals must have random partials generally decreasing with the order  $n$ . However, as already remarked, signals with  $p = 0$  can represent ambiguous situations in which, for some particular reason, the tones have an unusual blend of partials in the first part of the Fourier series. In this case a certain jamming of the pitch sensation can be produced. More usual situations (associated with a more precise sensation of pitch) will arise if in some way we modulate the  $\phi_n$  in order to get the amplitudes. In the simulations presented here, we have adopted the choice (3.1) with  $p$  extended to every positive number, and the amplitudes of the partials of our fundamental frequency  $\nu_k$  will be simply defined as  $c_n = \phi_n/n^p$ .

Now, following the discussion of Section 2, we can attribute the following weights to the 12 relative locations in the diatonic scale (starting with  $\nu_k$ )

$$\begin{aligned}
b'_0 &= \sqrt{c_1^2 + c_2^2 + c_4^2 + c_8^2 + c_{16}^2} & b'_6 &= \sqrt{c_{11}^2 + c_{22}^2 + c_{23}^2} \\
b'_1 &= c_{17} & b'_7 &= \sqrt{c_3^2 + c_6^2 + c_{12}^2 + c_{24}^2} \\
b'_2 &= \sqrt{c_9^2 + c_{18}^2} & b'_8 &= \sqrt{c_{13}^2 + c_{25}^2 + c_{26}^2} \\
b'_3 &= c_{19} & b'_9 &= c_{27} \\
b'_4 &= \sqrt{c_5^2 + c_{10}^2 + c_{20}^2} & b'_{10} &= \sqrt{c_7^2 + c_{14}^2 + c_{28}^2 + c_{29}^2} \\
b'_5 &= c_{21} & b'_{11} &= \sqrt{c_{15}^2 + c_{30}^2 + c_{31}^2}.
\end{aligned}$$

Then the  $b'_j$  will be rearranged in order to attribute the suitable weights to the 12 pitch classes. The pitch class of the fundamental  $\nu_k$  will be  $\nu_{l_k}$  with  $l_k = k \bmod 12$ : to this pitch class we will attribute the weight  $b_{l_k} = b'_0$ ; to the pitch class  $b_{l_k+1}$  we will attribute  $b_{l_k+1} = b'_1$  and so on. Finally, in order to make the entire simulation independent from the intensity of the signals, we will normalize the templates of the  $b$ 's by defining

$$x_k = \frac{b_k}{N}; \quad N = \sqrt{\sum_{j=0}^{11} b_j^2}. \quad (3.2)$$

In order to produce random signals without a detectable pitch, we need only to take a sample of 120 random variables (uniformly distributed) in  $[0, 1]$ , let us say  $\phi_n$ ;  $n \in \{-60, \dots, 59\}$ , which will represent the amplitudes of the Fourier transform attributed to every  $\nu_k$ . Then we calculate

$$b_k = \sqrt{\sum_{j=-5}^4 \phi_{k-12j}^2}; \quad k \in \{0, 1, \dots, 11\},$$

and finally use (3.2). Of course this procedure produces templates corresponding to *white* noise. If we want *coloured* noise it will be enough to introduce a bandpass filter  $H_n$  with values 1 (if the

frequency  $\nu_n$  passes) and 0 (if  $\nu_n$  is absorbed). In this case we first of all calculate  $c_n = H_n\phi_n$ , then

$$b_k = \sqrt{\sum_{j=-5}^4 c_{k-12j}^2}; \quad k \in \{0, 1, \dots, 11\},$$

and finally use again (3.2).

Besides these samples of signals with or without a detectable pitch we will need to produce some other type of signals to test the performances and the generalization power of this model. In particular, in the following simulations, we will utilize two other types of inputs:

- a) signals with a *residue pitch*;
- b) signals produced by means of a *superposition* of two or more signals.

The first type of signals consists of periodic signals constituted only by means of their high harmonics; namely they are represented by Fourier series where the first (and more prominent) harmonics are filtered away. It is well known that, even if the fundamental frequency is no longer present, the perception of the same pitch persists and it is known as *residue pitch* since it is extracted from a residue of high harmonics (Hall D. E. 1982; Zwicker E. and Fastl H. 1990). In the simulations these signals will simply be produced like the other signals with a pitch, but the value of  $c_1$  (and of  $c_2$  in further simulations) will automatically be zero. The second type of signals will reproduce intervals or chords and they will be simulated by means of the superposition of two (or more) signals with detectable pitch; namely we will sum the squares of the weights of the same pitch classes in the two signals, and we will attribute the square root of this sum as weight of this pitch class in the total signal. Of course the final template will also be normalized.

**3.2 Training and Performance:** In the two series of simulations implemented we used neural networks with 12 input neurons (in order to be adapted to our random templates with 12 entries as inputs), and just 1 output neuron with an activation value which ranges continuously between 0 and 1: if the output is 1 we intend that a pitch has been identified (we do not specify *which one* at this stage of the research); if, on the contrary, the output is 0, no pitch has been detected. Of course the values of the output can also indicate more ambiguous situations. The two series of simulations are different for the structure of the training set of examples and for the architecture of the neural network. On the other hand, the examples used in the verification and in the generalization have been built in the same way. More precisely the performances of the neural network have been always tested in two ways: first of all we verified if the system give the right outputs when the inputs are of the same type of the training set (namely signals with or without pitch simulated exactly as that of the training set). Then we tested the possible generalizations with input data of a type different from that of the training set (namely signals with residue pitch

and superpositions of signals).

The *first set of simulations* utilizes the simplest possible Neural Network following the results of the Kolmogorov theorem (Hecht-Nielsen R. 1989), namely a three-layers net with 12 input neurons, 1 output neuron and  $25 = 2 \cdot 12 + 1$  neurons in the hidden layer. The training set of examples is constituted of 500 samples drawn from our *random acoustic environment* in the following way: first of all there are 250 samples of periodic signals associated with the output 1 (since in the supervised training we suppose that they are recognized as endowed with a detectable pitch); they are produced with the  $c_n$  simulated as values of the 31 random variables

$$\xi_n = \frac{1}{n^p} \xi; \quad (p \geq 0; n = 1, 2, \dots, 31)$$

where  $\xi$  is uniformly distributed in  $[0, 1]$ . More precisely there will be 50 samples with  $p = 0$ , 50 with  $p = 0.5$ , 100 with  $p = 1$  and 50 with  $p = 2$ . This means that we will include in the training set even some noisy tones with a pitch and that we will suppose that they are perceived as endowed with a detectable pitch. The remaining 250 samples are typical noises simulated as previously described: 100 samples are *white* noises associated with a required answer 0 (since we suppose that in a broad band noise there will be no detectable pitch); then there are samples of *coloured* noise, namely 50 with a bandpass  $[\nu_{-20}, \nu_{20}]$  with required output 0.1, and 100 with bandpass  $[\nu_1, \nu_{12}]$  with required output 0.5 (since we suppose that when we narrow the bandpass we can elicit some pitch perception: in fact we select a part of the frequency axis with a more or less defined height). We remark here that, in order to have good performances from the net, the number of the examples of the training set must be at least equal to the total number of connection and threshold parameters of the net. In our case we have  $500 > 12 \cdot 25 + 25 + (12 + 25 + 1) = 363$ . By summarizing the training set of the first simulation is:

<i>type of examples</i>	<i>quantity</i>	<i>p or bandwidth</i>	<i>required output</i>
harmonic tones	50	0	1
harmonic tones	50	0.5	1
harmonic tones	100	1	1
harmonic tones	50	2	1
noises	100	white	0
noises	50	$[\nu_{-20}, \nu_{20}]$	0.1
noises	50	$[\nu_1, \nu_{12}]$	0.5

**Tab. 3.1**

After 5,500 iteration of the backpropagation algorithm the neural network learns to reproduce exactly 486 of the 500 examples with a *total* mean square error of about 1.48. At this point the neural network is ready to be tested; in order to do that (these remarks are suitable also for the

second series of simulations) we produce 20,000 samples of every particular type of signal and we consider the histograms of all the corresponding outputs (which of course are all numbers in  $[0, 1]$ ). In all our simulations the results will be given with a confidence coefficient of 95%, in the sense that the intervals indicated in the Figures will contain the true value of  $q$  (namely of the probability that the output will fall in one of the subintervals of  $[0, 1]$ ) with probability  $1 - \alpha = 0.95$  (see Appendix for details).

The results of this first round of simulations are given in the Figures at the end of the paper: there the outputs are listed respectively for harmonic tones (with different values of  $p$ : Fig. **I.1**÷**4**), for noises (with different bandwidths: Fig. **I.5**÷**7**) and for tones with residue pitch (with different values of  $p$ : Fig. **I.8**÷**15**). As for the superposition of tones (intervals) instead of the histograms we have calculated a *pitch parameter* which is nothing else than the expectation value of the outputs for every class of intervals: it can be considered as a measure of how much the perception of a detectable pitch is disturbed or reinforced by the superposition of tones. In the simulations for tone superpositions we have always taken  $p = 1$ .

The *second set of simulations* has been implemented by means of a different neural architecture: the network has still 12 input and 1 output neurons, but the hidden layers are 2 and are composed respectively of 25 and 12 neurons. This modification will modify the performances of the net in some respects, as will be seen later. Since now the number of neurons is larger, even our set of training examples must be larger: we will consider a set of 1,000 examples, since  $1,000 > 12 \cdot 25 + 25 \cdot 12 + 12 + (12 + 25 + 12 + 1) = 662$ . Of course this will also slow the training phase: in fact we will need now some 10,000 iterations in order to have 931 examples perfectly reproduced and an overall mean square error of 1.97. Our 1,000 training examples are now produced in a way perfectly identical to that of the first simulations, but the proportions of the different type of signals and some of the required answers are different. More precisely we will have now

<i>type of examples</i>	<i>quantity</i>	<i>p or bandwidth</i>	<i>required output</i>
harmonic tones	200	0	0.7
harmonic tones	100	0.5	0.9
harmonic tones	100	1	1
harmonic tones	100	2	1
noises	200	white	0
noises	100	$[\nu_{-20}, \nu_{20}]$	0.1
noises	200	$[\nu_1, \nu_{12}]$	0.3

**Tab. 3.2**

The results are given in a way identical to that adopted for the first simulations in the Fig. **II.1**÷**16** at the end of the paper.

## 4. DISCUSSION AND CONCLUSIONS

Looking at the results listed in the Figures **I.1÷7** and **II.1÷7**, the first remark to do is that the performances of our neural networks are satisfactory as long as we are interested in signals of the *same type* of those of the training sets. In other words we can say that, in both the first and the second simulation, the network can discriminate between signals *with* and *without* detectable pitch. Of course the performances are optimal when the samples come from well separated sets of signals: for examples for  $p = 1$  and  $p = 2$  (Fig. **I.3÷4**; **II.3÷4**) the presence of a pitch is very clear (more than 90% of the outputs are between 0.9 and 1.0), and, on the other hand, for white noises (Fig. **I.5**; **II.5**) is the absence of a pitch to be evident (more than 90% of the outputs are between 0.0 and 0.1).

The performance can also be considered good in the intermediate cases: in fact the discrimination between noises and tones is now less clear, but that must be so as remarked in the previous sections. Indeed from the Fig. **I.1÷4** and **II.1÷4** we can see that a periodic signal becomes more noisy (in the sense that the pitch is less perceptible) when  $p$  becomes smaller; and, on the other hand, the more the bandpass of a noise is narrow, the more a pitch emerges from the signal (see Fig. **I.6÷7** and **II.6÷7**). However in these intermediate situations we must also remark the tendency of the outputs to be *polarized* toward 0 and 1 (with the exception of **II.7**): the intermediate values are almost never predominant and the ambiguous situations manifest themselves rather in the fact that the probabilities of the outputs 0 and 1 have similar values. Moreover this behavior is more prominent in the first set of simulations than in the second and a first explication can be the fact that, in the supervised training, the examples of the first simulations presented more polarized answers than the second: we will have more to say on that later. A first consequence of these remarks is that we can consider meaningful to use the reduced Fourier transform in our simulations since the sets of signals with and without pitch remain fairly separated also after the projection.

Let us now consider the results of our simulations when the inputs are no more of the same type of the training examples: namely, let us analyze the possible generalizations of the performances of the Network. We tested these generalization abilities in two ways: first of all we used as inputs periodic signals from which the first (or the first two) partial has been subtracted. Namely we tested the output with tones lacking the fundamental frequency (and possibly even the second partial). The aim of this simulations is to see if a neural network, trained to detect pitches from random signals (*with* fundamental frequency), can detect a *residue pitch* even when the first partials are filtered away from the tone. Of course what we have in mind is to verify if *the perception of a residue pitch can be simulated as a phenomenon of pattern recognition of an incomplete signal by*

a neural network trained to detect the pitch in complete signals.\* Given the fact that our model based on the reduced Fourier transform must be considered as a very crude approximation of the reality, we think very encouraging the fact that our trained network can detect a residue pitch from the signals modified by the subtraction of one or two partials (see Fig. **I.8÷15** and **II.8÷15**). In fact the results of these simulations show the same qualitative behaviour presented in the case of the complete signals, but for the fact that the detection of the residue pitch is less clear than the perception of the normal pitch (the value of the probability for outputs between 0.9 and 1.0 are less prominent) as happens also in the real world. Moreover, even here the detection of the residue pitch is more clear when the tone is not noisy, namely for larger values of  $p$ , and this is especially true for the second series of simulations (see Fig. **II.8÷9**). A consequence of this result is that we can in some sense consider that our simple model of the real acoustic world makes some sense since at least it allows one to give an account of some phenomena (the residue pitch perception) by means of simpler facts (the pitch perception of complete signals taken as an elementary fact).

More unexpected, on the contrary, are the results of the second type of generalization: we tested the net by means of signals obtained as superpositions of two periodic signals (as described in the Section 3) and we have calculated a *pitch parameter* for these signals. The aim was to see if the templates of these pitch parameters vs. the musical intervals would have reproduced the well known templates of the perceived *consonance* or *dissonance* (a first germ of the *tonal relations*) of these intervals<sup>†</sup>. However in this case it is very clear that the known relations between tones in intervals are not well reproduced. For example small values are given to Sixths ( $I - VI$ ) and Thirds ( $I - III$  and  $I - IIIb$ ) and on the contrary high values are attributed to intervals like Diminished Fifths ( $I - IV\sharp$ ) Seconds ( $I - II$ ) and Sevenths ( $I - VII$ ), despite the fact that Sixths and Thirds are considered more consonant than Seconds or Sevenths. Moreover this performance is not improved if we consider the outputs of the second neural architecture: the only difference between these two sets of results is in the fact that the histogram **II.16** is more *flat* than **I.16**, but this can be a consequence of both, the different architecture of the network and the different set of training examples.

At first sight we could deduce from these results that a system able to individuate the presence of a pitch in a signal can not distinguish between consonant and dissonant intervals. This could suggest that the perceptions of consonance and dissonance are not based on a system of (respectively) confirmation and denial of the presence of a fundamental pitch. In other words: the

---

\* This model of pattern recognition for the residue pitch can of course claim some resemblance with the well known ideas of Terhardt, Wightman and Goldstein (Zwicker E. and Fastl H. 1990; Hall D. E. 1982).

<sup>†</sup> See for example the templates of relationship between pitches measured by Krumhansl C. as reproduced in Leman M. 1990.

information about the presence of a pitch is not (qualitatively) the same as the information about the consonance and dissonance of combined signals. However these conclusions are apparently too sharp if we take into account the following remarks.

First of all we should reconsider the method for the attribution of the partials to the pitch classes. As already remarked some of these attributions are in fact ambiguous since the real frequency is more or less equidistant from two *well tempered* frequencies. In fact this results also in a not very balanced attribution of partials: for example there are four partials (even if of an high order) attributed to the Minor Sevenths, three to the Diminished Fifths and only one to the Minor Thirds (see Tab. 2.3). Hence it is possible that a more balanced situation can be obtained if we do not attribute the whole energy of every partial to just one pitch class, and instead we adopt a strategy of spreading this energy on a suitable interval attributing to every pitch class only the share of the signal energy falling in the appropriate interval. In order to do that perhaps the best idea is to remember that every frequency of a spectrum excite the nerves of a region of the basilar membrane whose width is about 1 mm. This well known fact gives rise to the phenomenon of the *critical bandwidth* (Hall D. E. 1982; Zwicker E. and Fastl H. 1990) whose range is of about  $15 \div 20\%$  of the center frequency, namely more or less  $2.5 \div 3$  semitones. That means that partials which are less than a Minor Third apart begin to overlap on the basilar membrane spreading their energy on more than one well tempered frequency. If we take this fact into account we will probably be able to design a different algorithm for the attribution of weights to the pitch classes, and that can of course change the results of our simulations. Moreover the use of an excessive number of partials can be considered in some sense as an addition of noise to the tone making too difficult the retrieval of a pitch. The proposed spreading of the signal energy over larger intervals can also make sure that no pitch class will have an exactly zero weight even if we take into account a smaller number partials.

The second remark to do is that we should also modify the way in which we superpose periodic signals. In fact, if we consider our discretized Fourier transforms as elements of a vector space with the basis given by the trigonometric functions with discretized frequencies, our two signals will be superposed by simply summing up the *components*  $c_n^{(1)}$  and  $c_n^{(2)}$  as  $c_n = c_n^{(1)} + c_n^{(2)}$  and then projecting the total signal in the corresponding reduced Fourier transform. In this way we will obtain a weight attributed, for instance, to the first pitch class of the form

$$\sqrt{(c_1^{(1)} + c_1^{(2)})^2 + (c_2^{(1)} + c_2^{(2)})^2 + \dots + (c_{16}^{(1)} + c_{16}^{(2)})^2}$$

instead of the weight (used in our simulations, as described in Section 3.2)

$$\sqrt{(c_1^{(1)})^2 + (c_1^{(2)})^2 + (c_2^{(1)})^2 + (c_2^{(2)})^2 + \dots + (c_{16}^{(1)})^2 + (c_{16}^{(2)})^2}.$$

This will of course modify the answers of the network.

Let us conclude this discussion with three more remarks. First of all a modification of the model which is very simple to implement is the fact that we can produce our samples by means of random variables which are not uniformly distributed: it is not clear that this will dramatically improve the results, but it is interesting to test how the performances change as a consequence of a change in the training set. From this standpoint the second remark, connected with the first, is that it will be important to test how the performances will change if the required outputs in the supervised training are modified. In fact it must be stressed here that it is not very well clear how to fix the required answers in the ambiguous cases. Many possible choices are possible and no evident motivations can be given for particular values of the outputs different from either 0 or 1. For example in our first set of simulations we decided to give output 1 to every periodic signal (even to the ambiguous noisy tones) while in the second set we gave less precise answers to the noisy tones. This resulted in performances which are less polarized on the extreme values 0 and 1, but the difference between the two sets of simulations is not particularly evident. In fact in the ambiguous cases the neural network tends to give as output either 0 or 1, but not the intermediate values (only in Fig. II.7 we got a *flat* histogram, and even in this case the intermediate values are not prominent). Of course these remarks once more stress the idea that it will be very interesting to perform the experiment by reducing *real* signals and by adopting *real* answers about the pitch. However in our opinion the problem is less in the suspicion that the simulation could artificially introduce a distinction between tones and noises on the basis of some regularity (the randomness of the sample production makes sure that this will not happen), than in the fact that the training set can be grossly unbalanced in the composition of the samples and of the answers.

The previous discussion introduces also our last remark: we stated in the Section 2 that a backpropagation neural network can be seen as a computing system which implements a good approximation (in the sense of the least squares) of the function  $E(Y|X = x)$ . However this is not so sure if the training set of examples is produced as in our simulations. Indeed our (normalized) reduced Fourier transform are the points of a 11-dimensional hypersphere embedded in  $R^{12}$ . When we give our random sample (of size  $N$ ) of pairs signal/output  $(x^{(k)}, y^{(k)})_{k=1, \dots, N}$ , with probability 1 we never get twice exactly the same point  $x$  on our 11-dimensional hypersphere and hence we never have the possibility to show to the learning neural network that *to the same point  $x$  different outputs  $y$  can be associated*. That means that, unless our sample is overwhelmingly populous (very large  $N$ ), the neural network will not always do a conditional average, but rather will try to follow the irregular oscillations of the associations given in our sample  $(x^{(k)}, y^{(k)})_{k=1, \dots, N}$  as far as the set of its possible functions will allow it. The fact that in our trainings more than 90% of the examples were exactly reproduced (486 out of 500 in the first simulation and 931 out of 1,000 in the second)

is in our opinion an indication of this problem. This is likely to be the main reason for the quoted phenomenon of the *polarization* of the outputs toward the extreme results 0 and 1 (instead of intermediate results) even when the samples used as tests are clearly ambiguous. Of course we can try to avoid this problem by giving a large number of training examples, but this does not seem to be the better way since the time of training grows very fast with the size of the training sample: for instance, in our two sets of simulations, the first network (363 parameters to be determined) completed the training on 500 examples in about 90 minutes of CPU time, while the second (662 parameters) with 1,000 examples required about 530 minutes. The best thing seems to us to be a modification either in the algorithms of the backpropagation neural network or in the presentation of the training samples in order to get a sort of *regularization* of the functions implemented by the trained network. For example we could manipulate the set of training examples in order to give to the network an indication of the fact that to the same input can correspond more than one output. A simple way to do that would be to fix a *regularization parameter*  $\rho$  (its magnitude must of course be initially estimated in some way), to consider around every input  $x^{(k)}$  an hypersphere of radius  $\rho$ , to check if in this hypersphere there are other points of our sample (presumably with a different output), and finally to attribute to  $x^{(k)}$  all the outputs of the points belonging to its spherical neighborhood. Of course this is just a first suggestion which need to be much more thoroughly investigated.

The results of this paper must be considered only preliminary and an assessment about the possibilities of our model will require further simulations and extensive modifications to be definitive. Hence we will finally indicate some of the directions in which this work could be extended. First of all an obvious modification will be to design networks able to detect not only *if* there is a pitch, but also *which* pitch class is present in the given signal. This will complicate a bit the network architecture, but in some preliminary attempts the classes of inputs seemed to be separated enough to be easily detected. A second modification will also be an attempt to take into account more elements of the complete Fourier transform of the signal: it is possible to take into account, for examples, the phases of the Fourier components, since it is well known that an influence of the phases on the perception of the pitch of complex tones exists (Plomp R. 1976). A third is to implement our analysis on real and not simulated signals.

A fourth point is the fact that, if we want to approach in a realistic way the question of the emergence of a tonal sensitivity, we must take into account the time evolution of the acoustic signals. In fact our study was just a *static* investigation, but everybody will agree that the tonal relations are something more complicated than just the presence of a detectable pitch or the relations of consonance between harmonic tones and that the time sequence of the signals will be a crucial element in this sort of analysis. However this remark points also to a last but not least question:

if we want to develop a *dynamical* analysis of acoustic signals we can not stick to our static model based on a backpropagation neural network. In this case a new model must be designed and we think that the appropriate approach will be that of the (deterministic or stochastic) dynamical systems endowed with autonomous attractors and driven by a superimposed external signal. Some work in this direction has already been started by other researchers (Leman M. 1992; D' Autilia R. and Guerra F. 1991).

Some other changes in the design of the model are also stimulated by the remark that the *supervised training* can not always be considered as the better way to deal with these problems. We already hinted to this question in the discussion of the results when we pointed out the fact that there is no clear motivation for the choice of a particular value of the output, apart from the extreme values 1 and 0 of the unambiguous signals. Of course this problem is even more relevant when we try to extend our simulations from the problem of detecting pitches to that of individuating chords and keys. In these complex cases is difficult to find a suitable representation for the outputs of our system. A solution to this problem could be to adopt a strategy of *non-supervised training*, for example by means of Kohonen maps, as already done in a different way by other researchers (Leman M. 1990).

**Acknowledgements:** The authors want to thank Prof. M. Leman and Prof. F. Guerra for invaluable discussions and suggestions.

## APPENDIX

In order to calculate our histograms for our 20,000 samples we subdivide the interval  $[0, 1]$  in 10 subintervals of amplitude 0.1, then for every subinterval we consider the (independent and identically distributed) random variables  $\eta_i \in \{0, 1\}$ ,  $i = 1, 2, \dots, 20,000$ :  $\eta_i = 1$  means that the  $i$ -th output falls in our subinterval, and  $\eta_i = 0$  means the opposite. Now we subdivide the  $\eta_i$  in 20 subsets of 1,000 samples each and we consider the sample averages

$$S_n = \frac{1}{1000} \sum_{k=1000(n-1)+1}^{2000(n-1)} \eta_k, \quad n = 1, 2, \dots, 20.$$

On the basis of the Central Limit Theorem, if  $q = P(\eta_i = 1)$ ;  $i = 1, 2, \dots, 20,000$ , we can say that these averages are all approximately normally distributed as  $S_n \sim \mathcal{N}\left(q, \frac{q(1-q)}{1000}\right)$ . Since we do not know the value of  $q$ , we will consider  $(S_n)_{n=1, \dots, 20}$  as a random sample of size 20 of a normal random variable with unknown expectation value  $\mu = q$  and variance  $\sigma^2 = q(1-q)/1000$ . Hence, if

$$\begin{aligned} \bar{S} &= \frac{1}{20} \sum_{n=1}^{20} S_n \\ \hat{\sigma}^2 &= \frac{1}{19} \sum_{n=1}^{20} (S_n - \bar{S})^2 \end{aligned}$$

are the unbiased estimators of the expectation value and of the variance, we know that

$$T = \frac{\bar{S} - \mu}{\hat{\sigma}} \sqrt{20}$$

is a *Student* random variable with 19 degrees of freedom so that we can calculate the suitable confidence intervals for the value of  $q$  which of course in our histogram represents the probability that the output of our neural network will fall in the chosen subinterval of  $[0, 1]$ . In all our simulations the results will be given with a confidence coefficient of 95%, in the sense that the intervals indicated in the figures will contain the true value of  $q$  with probability  $1 - \alpha = 0.95$ . More precisely the confidence interval will have the form

$$\left[ \bar{S} - \frac{\hat{\sigma}}{\sqrt{20}} t_{19, 0.975}, \bar{S} + \frac{\hat{\sigma}}{\sqrt{20}} t_{19, 0.975} \right]$$

where, from the values of the Student distribution,  $t_{19, 0.975} \simeq 2.093$ , and our results will be given as  $\bar{S} \pm \frac{\hat{\sigma}}{\sqrt{20}} t_{19, 0.975}$ .

## REFERENCES

- Courant R. and Hilbert D. (1953):** *Methods of Mathematical Physics*, Vol. I and II; Wiley, New York.
- D' Autilia R. and Guerra F. (1991):** *Qualitative Aspects of Signal Processing through Dynamical Neural Networks*; in *Representation of Musical Signals*, De Poli G, Piccialli A and Roads C Eds., M.I.T. Press, Cambridge.
- Hall D. E. (1982):** *Musical Acoustics: An Introduction*; Wadsworth, Belmont.
- Hecht-Nielsen R. (1989):** *Neurocomputing*; Addison-Wesley, Reading.
- Leman M. (1990):** *Emergent Properties of Tonality Functions by Self-Organization*; *Interface* 19, 85.
- Leman M. (1992):** *The recognition of Tone Centers in Acoustical Representation of Music*; Report SM-IPEM 24, University of Ghent.
- Poggio T. and Girosi F. (1989):** *A Theory of Networks for Approximation and Learning*; M.I.T. Artificial Intelligence Memo No. 1140.
- Plomp R. (1976):** *Aspects of Tone Sensation*; Academic Press, London.
- Rumelhart D. E. and McClelland J. L. (1986):** *Parallel Distributed Processing*; Vol. I and II; M.I.T. Press, Cambridge.
- Shiryayev A. N. (1984):** *Probability*; Springer-Verlag, New York.
- Zwicker E. and Fastl H. (1990):** *Psychoacoustics; Facts and Models*; Springer-Verlag, Berlin.