

# NIH Public Access

Author Manuscript

*leural Comput*. Author manuscript; available in PMC 2012 November 13.

# Published in final edited form as:

Neural Comput. 2005 April; 17(4): 881–902. doi:10.1162/0899766053429408.

# Rate and Synchrony in Feedforward Networks of Coincidence Detectors: Analytical Solution

#### Shawn Mikula and Ernst Niebur

Zanvyl Krieger Mind/Brain Institute and Department of Neuroscience, Johns Hopkins University, Baltimore, MD 21218, U.S.A.

# Abstract

We provide an analytical recurrent solution for the firing rates and cross-correlations of feedforward networks with arbitrary connectivity, excitatory or inhibitory, in response to steadystate spiking input to all neurons in the first network layer. Connections can go between any two layers as long as no loops are produced. Mean firing rates and pairwise cross-correlations of all input neurons can be chosen individually. We apply this method to study the propagation of rate and synchrony information through sample networks to address the current debate regarding the efficacy of rate codes versus temporal codes.

Our results from applying the network solution to several examples support the following conclusions: (1) differential propagation efficacy of rate and synchrony to higher layers of a feedforward network is dependent on both network and input parameters, and (2) previous modeling and simulation studies exclusively supporting either rate or temporal coding must be reconsidered within the limited range of network and input parameters used. Our exact, analytical solution for feedforward networks of coincidence detectors should prove useful for further elucidating the efficacy and differential roles of rate and temporal codes in terms of different network and input parameter ranges.

# **1** Introduction

Much of the theoretical basis for understanding information processing in complex biological systems is based on computational modeling-numerical solutions of the underlying model equations. While this approach has proven extremely useful and is the only practical one in many cases, analytical solutions would nearly always be preferable if they were available. Naturally, this is the case only for carefully selected systems of sufficient simplicity. In this report, we present one such system: an interconnected network of ideal coincidence detectors of arbitrary depth. Our solution is limited to feedforward networks, but otherwise the connectivity is arbitrary:neuronal thresholds can be arbitrary. synapses can be of arbitrary strengths, they can be excitatory and inhibitory, and they can be between any two neurons in any two layers provided that no loops are formed (this is the feedforward condition). The input to the network is characterized in terms of the average firing rate and the pairwise correlation, and we present iterative, closed-form solutions for all neurons (and pairs) in the network in the same form. The model is based on our previous exact, analytical solutions for the output firing rate of an individual coincidence detector receiving excitatory and inhibitory inputs, in both the presence and absence of synaptic depression (Mikula & Niebur, 2003a, 2003b, 2004)

<sup>© 2005</sup> Massachusetts Institute of Technology mikula@jhu.edu, niebur@jhu.edu.

After defining methods and notations in section 2, we present our main result, the closed-form expressions for mean rate and cross-correlation, in section 3. Example networks are studied in sections 4 and 5. Implications for neural coding are discussed in section 6.

# 2 Methods

#### 2.1 Notational Conventions

Matrix notation is used throughout our derivation. Matrices are denoted by boldfaced, uppercase letters; row and column vectors by boldface, lowercase letters; and scalars by regular lower-case letters. Matrices and vectors that are functions of one or more variables have their arguments denoted by subscripts, whereas scalars have their arguments within

parentheses. Thus, for instance,  $\vec{\Psi}_j$  denotes a matrix that is a function of one argument, *j*; the vector  $\vec{c}_{i,j}$  is a function of two arguments, *i* and *j*; and *q*(*i*, *j*; *s*, *t*) denotes a scalar that is a function of four arguments, *i*, *j*, *s*, and *t*.

#### 2.2 Model Neurons: Coincidence Detectors

The model neurons used in this study are coincidence detectors. A coincidence detector is a computational unit that fires at time *t* if the number of unit excitatory postsynaptic potentials (EPSPs) received within the window (t - T, t) equals or exceeds the threshold  $\theta$ . In many cases, it makes sense to think of *T* as a period on the order of 10 ms. This is the timescale of fast ionic synaptic conductances, and it is at this timescale that synaptic events superpose and interact. We do not, however, make use of this specific setting in our analysis, other than requiring that it be sufficiently small so that a maximum of one spike can be generated in a period of this length, and our results do not depend on this setting.

#### 2.3 Binomial Spike Trains with Specific Cross-Correlation

In a previous report Mikula & Niebur, 2003a), we introduced a systematic method for the generation of an arbitrary number of spike trains with specified pair-wise mean cross-correlations and firing rates. Action potentials are distributed according to binomial counting statistics in each spike train. Mean firing rates and cross-correlations are the same for all spike trains (or all pairs of spike trains, respectively), but they can be chosen independently of each other. We describe the procedure here for the convenience of the reader.

Let *m* be the number of input spike trains, each having *n* time bins. All bins are of equal length  $\delta t$ , chosen sufficiently small so that each contains a maximum of one spike; that is, each bin is guaranteed to contain either one or zero spikes. The makes the decision whether to fire within one time bin; therefore,  $T = \delta t$ . Assuming a firing rate of

$$f = \frac{p}{\delta t},$$
 (2.1)

the probability that a spike is found in any given time bin is p; no spike is found with probability(1 - p). Bins in any given spike train are independent, which implies that the following analysis can be limited to a single time bin. A physiologically important special case is obtained if the rate of incoming spikes is low and convergence is high; the binomial statistics that governs the spikes generated by a coincidence detector is then approximated by Poisson statistics. We further note that throughout this letter, we often refer to the spike probability, p, simply as an input or output firing rate, with the understanding that the actual firing rate is obtained by dividing p by the time bin size,  $\delta t$ , as in equation 2.1.

#### 2.4 Network Architecture

An example feedforward network is shown in Figure 1. To characterize the connectivity

between layers in the network, let us introduce the connectivity matrix, denoted  $\vec{C}_j$ , which quantifies the connectivity from the *j* layer to the *j* + 1th layer, and in which the (*k*, *l*)th entry contains the numerical value of the connection to the *k*th neuron in layer *j* + 1 from the *k*th

neuron in layer *j*. The size of  $\vec{C}_j$  is  $m_{j+1} \times m_j$ , where  $m_j$  denotes the number of neurons in layer *j*. The values of the connectivity matrix are real numbers—positive for excitatory connections and negative for inhibitoryconnections.

We define the connectivity vector, a row vector, denoted  $\vec{c}_{i,j}$ , to be the *i*th row from  $\vec{C}_{j}$ . It

follows from the definition of  $\vec{c}_j$  that the length of  $\vec{c}_{i,j}$  is  $m_j$  and that the *k*th element of  $\vec{c}_{i,j}$  contains a real number describing the connection from the *k*th neuron in layer *j* to the *k*th neuron in layer j + 1.

Although the example network in Figure 1 is limited to connections between subsequent layers, that is, between layers j and (j + 1) for all j, we note that our formalism is sufficiently general to allow connections between any two layers j and j + J with J = 1, provided layer j + J exists in the network. This can be seen by first considering a network with connections only between subsequent layers (as in Figure 1), that is, with J = 1. The connections with J > 1 are then added by formally adding virtual neurons with one incoming synapse whose strength exceeds its threshold.

Finally, we denote the output firing probability(which, by equation 2.1, is directly proportional to output firing rate) for the *i*th neuron located in the *j*th layer by the scalar firing rate, p(i, j).

#### **3 Results**

#### 3.1 Computation of Firing Rates for the Network

As discussed in section 2.3, the  $m_0$  initial inputs to our feedforward network are binomial and originate at the zeroth layer (see Figure 1). Because each of the  $m_0$  inputs is the outcome of a Bernoulli trial, the total number of possible input states is  $2^{m_0}$ .<sup>1</sup> We will find it useful to enumerate these  $2^{m_0}$  input states and to assign corresponding probabilities for their occurrence.

Toward this end, we define the input state matrix,  $\vec{\Psi}_0$ . Each row of this matrix represents one input state in the form of a binary row vector whose *k*th entry consists of a zero or a one, denoting the absence or presence of an input spike in the *k*th input, respectively. From the above considerations, we know that  $\vec{\Psi}_0$  must be of size  $2^{m0} \times m_0$ .

We now determine how the different input states are transformed at layer 1 of our network. That is, we compute how each input state affects suprathreshold events for all of the  $m_1$  neurons in layer 1. The solution is based on the known connectivities between the  $m_0$  inputs and the  $m_1$  neurons in layer 1,  $\overrightarrow{C}_0$  (see section 2.4), and on the thresholds of the neurons in layer 1, listed in the matrix  $\overrightarrow{\theta}_1$ . This matrix has  $2^{m_0}$  identical rows, each consisting of the

<sup>&</sup>lt;sup>1</sup>A Bernoulli trial is defined as a single random event for which there are two and only two possible outcomes that are mutually exclusive and have a priori fixed probabilities that sum to unity. In our case, the outcome of a trial will be either zero or exactly one spike of an input neuron in a given time bin; the former happens with probability p where 0 p 1 and the latter with probability (1 - p).

vector of all thresholds in layer 1; thus, the element in column *j* (and all rows) of this matrix is the threshold of neuron *j* in layer 1 (the usefulness of this construct will become obvious in equation 3.1). These network dynamics transform the input states, contained in the matrix  $\vec{\Psi}_0$ , into the resulting states in layer 1 of the network. We collect these states in the matrix

 $\vec{\Psi}_{1}$ , which describes the states of layer 1 and is computed as follows:

$$\vec{\Psi}_1 = \Theta\left(\vec{\Psi}_0 \vec{C}_0 - \vec{\theta}_1\right). \quad (3.1)$$

By the definition of matrix products, the element in row *i* and column j of  $\vec{\Psi}_0 \vec{C}_0$  is the input to neuron *j* in layer 1 when state *i* is present in layer 0.

Furthermore, we define the operator  $\Theta()$  in this equation as the Heaviside function acting on the individual entries of its matrix argument. It takes on the value of unity if an entry within its argument is nonnegative and zero otherwise.<sup>2</sup> Note that the dimensions of  $\vec{\Psi}_1$  are  $2^{m0} \times m_1$  and that the *k*th row of  $\vec{\Psi}_1$  is the layer 1 state corresponding to the *k*th row of  $\vec{\Psi}_0$ . That is, the transformation of the *k*th input state at layer 1 is given by the *k*th row of  $\vec{\Psi}_1$ . To be sure, while  $\vec{\Psi}_0$  is an exhaustive list of all states in the input layer,  $\vec{\Psi}_1$  is, in general, not a matrix containing all states in layer 1. Depending on  $m_0$ ,  $m_1$  and the connectivity and thresholds, this matrix in general contains only a subset of all possible states in layer 1, and each of these states may appear more than once.

We can recursively apply the same method used in equation 3.1 to generate the state

matrices for other layers of our network to arrive at the general expression for  $\vec{\Psi}_j$ , the layer *j* state matrix, as the following:

$$\vec{\Psi}_{j} = \Theta\left(\vec{\Psi}_{j-1}\vec{C}_{j-1} - \vec{\theta}_{j}\right), \quad (3.2)$$

where the *k*th row of  $\vec{\Psi}_j$  gives us the layer *j* state corresponding to the *k*th row of  $\vec{\Psi}_0$ . The iteration defines  $\vec{\Psi}_j$  in terms of  $\vec{\Psi}_{j-1}$  and  $\vec{C}_{j-1}$  and terminates at  $\vec{\Psi}_0$  and  $\vec{C}_0$ , as in equation 3.1.

The *i*th column of  $\vec{\Psi}_{j}$ , which is of length  $2^{m0}$ , gives us the neuronal responses of the *i*th neuron in layer *j* to the  $2^{m0}$  input states. Let this column vector be denoted  $\vec{\psi}_{i,j}$ , the state vector for the *i*th neuron in layer *j*; then the following holds:

$$\overrightarrow{\psi}_{i,j} = \Theta\left(\overrightarrow{\Psi}_{j-1}\overrightarrow{c}_{i,j-1} - \theta\left(i,j\right)\right), \quad (3.3)$$

where the scalar  $\theta(i, j)$  is the threshold of neuron *i* in layer *j*. Note that the only difference between the right-hand sides of equations 3.3 and 3.2 is that the latter substitutes the connectivity matrix  $\vec{C}_{j-1}$  and the threshold matrix  $\vec{\theta}_j$  for the connectivity vector  $\vec{c}_{i,j-1}$  and the threshold scalar  $\theta(i, j)$ , respectively.

 $<sup>^{2}</sup>$ We note that this component-wise procedure is not an operation defined in a vector space. We nevertheless use the terminology of vector spaces (rather than introducing terms like *arrays* or *n*-tuples) for convenience.

Neural Comput. Author manuscript; available in PMC 2012 November 13.

Having determined how input states are transformed across the network by equations 3.1 and 3.2, we now proceed to compute mean output spiking probabilities. We define the input state probability vector  $\vec{p}_{in}$ , a column vector whose *k*th element is the probability for the occurrence of the *k*th input state. The output spiking probability for any neuron in the network is obtained by summing over all input states, leading to suprathreshold events times the corresponding input state probability (given by  $\vec{p}_{in}$ ). Thus, the output spiking probabilities for all neurons located in layer *j* can then be written as a column vector  $\vec{p}_{j}$ , computed as

$$\overrightarrow{p}_{j} = \overrightarrow{\Psi}_{j}^{T} \overrightarrow{p}_{in}, \quad (3.4)$$

where the *i*th entry of  $\overline{P}_j$  is the output spiking probability for the *i*th neuron located in layer *j* and where  $\overrightarrow{\Psi}_j^T$  is the transpose of  $\overrightarrow{\Psi}_j$ .

Alternatively, by using equation 3.3, we can obtain an explicit expression for the output spiking probability of a single neuron located anywhere in the network. The output spiking probability for the *i*th neuron located in layer *j*, the scalar p(i, j), is given by

$$p(i, j) = \overrightarrow{\psi}_{i,j}^T \overrightarrow{p}_{in}.$$
 (3.5)

Note that p(i, j) is the same as the *i*th entry of  $\overrightarrow{p}_i$  from equation 3.4.

With equations 3.4 and 3.5, we have thus obtained an analytical recurrent solution for the output spiking probabilities (or firing rates per equation 2.1) of the coincidence detectors comprising a feedforward network that receives an arbitrary number of binomial inputs modulated with respect to both mean rate and cross-correlation.

#### 3.2 Computation of Cross-Correlations

The standard Pearson correlation coefficient, denoted q(x, y), is defined by the following for random variables x and y,

$$q(x, y) = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sqrt{\operatorname{var}(x)} \sqrt{\operatorname{var}(y)}}, \quad (3.6)$$

where  $\langle x \rangle$  denotes the mean value of x, and var(x) denotes the variance of x. If x is a Bernoulli random variable (i.e., takes on values 0 or 1), then its variance is given by

$$\operatorname{var}(x) = \langle x \rangle - \langle x \rangle^2, \quad (3.7)$$

and we may rewrite equation 3.6 as the following (Mikula & Niebur, 2003b):

$$q(x, y) = \frac{\Sigma \operatorname{prob} (xy=1) - \Sigma \operatorname{prob} (x=1) \Sigma \operatorname{prob} (y=1)}{\sqrt{\Sigma \operatorname{prob} (x=1) - (\Sigma \operatorname{prob} (x=1))^2} \sqrt{\Sigma \operatorname{prob} (y=1) - \Sigma (\operatorname{prob} (y=1))^2}}$$
(3.8)

where prob(x=1) denotes the probability that *x* takes the value of unity, and prob(xy=1) denotes the probability that both *x* and *y* take the value of unity.

In the notation established earlier, the spiking probability scalar p(i, j) of neuron j in layer i takes the place of  $\Sigma$ prob(x=1) in equation 3.8, taking into account that it is based on a

Bernoulli process. In this equation, to compute the correlation with neuron *t* in layer *s*, the probability p(s, t) takes the place of  $\Sigma \text{prob}(y=1)$ . And finally, since the joint probability for the *t*h neuron in layer *j* and the *s*th neuron in layer *t* to fire action potentials is given by

 $\Theta\left(\overrightarrow{\psi}_{i,j} + \overrightarrow{\psi}_{s,t} - 2\right)^T \overrightarrow{p}_{in}, \text{ as can be seen by direct substitution, it follows that } \Sigma \text{prob}(xy=1) \text{ is }$ 

equivalent to  $\Theta(\vec{\psi}_{i,j} + \vec{\psi}_{s,t} - 2)^T \vec{p}_{in}$ . We can therefore rewrite equation 3.8 as the following expression for the cross-correlation, denoted q(i, j; s, t), between the *i*th neuron in layer *j* and the *s*th neuron in layer *t*,

$$q(i, j; s, t) = \frac{\Theta(\vec{\psi}_{i,j} + \vec{\psi}_{s,t} - 2)^T \vec{p}_{in} - p(i, j) p(s, t)}{\sqrt{p(i, j) - p(i, j)^2} \sqrt{p(s, t) - p(s, t)^2}},$$
(3.9)

where  $\vec{\psi}_{i,j}$  is obtained using equation 3.3.

We have thus obtained an exact recurrent solution for the cross-correlation between any two coincidence detectors, or between any coincidence detector and an input, located within a feedforward network receiving an arbitrary number of binomial inputs modulated with respect to both mean rate and cross-correlation.

#### 3.3 Large Networks

On a practical note, we find that it is useful to consider methods for reducing the size of  $\overline{\Psi}_0$  for the purpose of making our solutions for rate and cross-correlation applicable to larger

feedforward networks. As we saw in section 3.1, for  $m_0$  inputs, the size of  $\vec{\Psi}_0$  is  $2^{m0} \times m_0$ , which increases quickly for increasing  $m_0$ . We assume that the connectivity of the network is given. Under this condition, we have discerned three practical methods for reducing the

size of  $\vec{\Psi}_0$ . One of them is to include only suprathreshold events, which means excluding those input states in which a subthreshold number of spikes occurs; all of these are mapped on a single state with a zero everywhere. Assuming uniform thresholds for all neurons,

which we will denote by  $\theta$ , then this method reduces  $\vec{\Psi}_0$  to size  $\left(2^{m0} - \sum_{j=0}^{\theta-1} {m_0 \choose j}\right) \times m_0$ , which can be a significant reduction as  $\theta/m_0$  approaches unity. The mirror image of this simplification is obtained in the case of small  $\theta/m_0$ ; in this case, a large number of input states is mapped onto one state, namely, "one" in all positions. The second method for

reducing the size of  $\vec{\Psi}_0$  is valid for low input rates (i.e., the Poisson regime), in which case the input states characterized by many spikes can be ignored due to the unlikely probability of such a state. The third method is to reduce the number of inputs while maintaining a relatively large number of neurons in each layer. In many cases, this is justified since for large convergence (the physiologically relevant case), only a small number of neurons can be simultaneously active (within one time bin of length *T*) to avoid the need for unrealistically high thresholds or saturation of the next layer.

## 4 Simple Example

Let us consider the simple *n*-layer excitatory feedforward network shown in Figure 1 with four neurons in each layer, cyclical (periodic) boundary connectivity conditions, and  $\theta(i, j) = \theta = 2 \forall i, j$ . We further assume that each connection is excitatory and has a weight of +1. The connectivity patterns between layers are the same for all layers; that is, each neuron in layer *j* connects to the three closest neurons in layer *j* + 1. Recalling from section 2.4 that the (*k*, *l*)th entry of the connectivity matrix is defined as the weight of the connection to the *k*th

neuron in layer j + 1 from the *I*th neuron in layer *j*, we obtain a connectivity matrix that is the same for each layer and given by the following:

$$\vec{C}(j) = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$
 (4.1)

For computing the total number of input states, we note that there are four binomial inputs, and thus  $2^4$  input states. The resulting input matrix  $\vec{\Psi}_0$  is

$$\vec{\Psi}_{0} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \quad (4.2)$$

Thus, the first row of  $\vec{\Psi}_0$  is the input state for having zero input spikes for all four inputs, the second row is the input state for having zero input spikes for inputs 1 through 3, whereas input 4 has a spike, and so on for all of the 16 rows of  $\vec{\Psi}_0$ .

Using the network connectivity and the input state matrices, we can now apply either equation 3.2 or 3.3 to derive  $\vec{\Psi}_j$  and  $\vec{\psi}_{i,j}$ . The result is given in Table 1, which shows all attainable states for all neurons in layers 1 through 3 of our network (plus the input layer, 0). We have placed  $\vec{\Psi}_0$ ,  $\vec{\Psi}_1$ ,  $\vec{\Psi}_2$ , and  $\vec{\Psi}_3$  alongside each other in Table 1 to emphasize that this is a logical truth table. This should also make it easier to see how different input states, which correspond to the rows of  $\vec{\Psi}_0$ , are transformed across different layers, which correspond to the rows of  $\vec{\Psi}_1$ ,  $\vec{\Psi}_2$ , and  $\vec{\Psi}_3$ .

It turns out that Table 1 characterizes the behavior not only of the first three but of all layers of the (infinite) network. The reason is that all higher even layers are identical to  $\vec{\Psi}_2$ , and similarly, all higher odd layers are identical to  $\vec{\Psi}_1$ . This periodic behavior is not a peculiarity of this specific network, is a property of all feedforward networks of coincidence detectors with identical layer properties (connectivity and thresholds) and finite numbers of neurons per layer; since they can have only a finite number of states, networks of infinite depth need to show periodic behavior. This can be seen easily by considering that there are  $2^{m0}$  possible states in the input layer, layer 0 (see section 3.1), and in all higher layers. Since all layers are

assumed to have the same number of neurons, at least two layers between the input layer and layer  $(2^{m0} + 1)$  must have identical activity patterns. Let us assume the first one of these is layer *I* and the second one is layer *I*. Then, since the connectivities between all layers are identical, the pattern in layer (I + 1) is the same as in (I + 1), the pattern in (I + 2) is as in (I + 2), and so on; activity is thus periodic with a periodicity (I + I) starting at layer *I*. This is what was to be shown. We note in passing that periodicity includes the case of period 1, for example, when the activity "dies out" in layer *I* because all activity is below threshold and therefore all higher layers have zero activity.

The next step is the computation of column vector  $\overrightarrow{p}_{in}$ , whose *i*th entry contains the probability for the *i*th input state (i.e., the *i*th row of  $\overrightarrow{\Psi}_0$ ). We will first assume uncorrelated inputs (correlated input is discussed later; see equation 4.6 for an example). In this case, the

components of  $\overrightarrow{p}_{in}$  are simple binomial probabilities. We have the following for  $\overrightarrow{p}_{in}$ :

$$\vec{p}_{in} = \begin{pmatrix} (1 - P_{10}) (1 - P_{20}) (1 - P_{30}) (1 - P_{40}) \\ (1 - P_{10}) (1 - P_{20}) (1 - P_{30}) P_{40} \\ (1 - P_{10}) (1 - P_{20}) P_{30} (1 - P_{40}) \\ (1 - P_{10}) (1 - P_{20}) P_{30} P_{40} \\ (1 - P_{10}) P_{20} (1 - P_{30}) (1 - P_{40}) \\ (1 - P_{20}) P_{20} P_{30} (1 - P_{40}) \\ (1 - P_{20}) P_{20} P_{30} (1 - P_{40}) \\ P_{10} (1 - P_{20}) (1 - P_{30}) (1 - P_{40}) \\ P_{10} (1 - P_{20}) (1 - P_{30}) (1 - P_{40}) \\ P_{10} (1 - P_{20}) P_{30} P_{40} \\ P_{10} (1 - P_{20}) P_{30} P_{40} \\ P_{10} (1 - P_{20}) P_{30} P_{40} \\ P_{10} P_{20} (1 - P_{30}) (1 - P_{40}) \\ P_{10} P_{20} (1 - P_{30}) (1 - P_{40}) \\ P_{10} P_{20} (1 - P_{30}) (1 - P_{40}) \\ P_{10} P_{20} P_{30} (1 - P_{40}) \\ P_{10} P_{20} P_{30} (1 - P_{40}) \\ P_{10} P_{20} P_{30} P_{40} \\ \end{pmatrix}$$

To alleviate the notation, we replaced p(i, 0) by  $P_{i0}$  for  $i \in \{1, ..., 4\}$ , that is, the mean firing probabilities for the *i*th input from the zeroth layer. The values of  $P_{i}$  can be freely chosen depending on the question under study (subject to the conditions 0  $P_{t0}$  1 since they are probabilities). For example, we may choose uniform inputs, in which case all  $P_{A}$  would be equal. Alternatively, we may study how spatially localized high rate inputs are dissipated in higher levels of the network. In this case, we set some of the  $P_{i0}$  to high values relative to the remaining  $P_{i0}$ . Both cases are illustrated in the examples. The results for the case of uncorrelated inputs of uniform rate are shown in Figure 2. Figure 2A shows the mean rates for the input, even, and odd layers. Note that rates get propagated to arbitrarily high levels of the feedforward network. Besides the initial attenuation of rates at layer 1, there is no further attenuation. Figure 2B shows nearest-neighbor cross-correlations for the network, and Figure 2C shows cross-correlation matrices for each layer. As expected, because we have uncorrelated inputs of uniform rate, nothing much interesting happens with the crosscorrelation at higher layers, except for a uniform increase in all cross-correlations, which is expected due to shared connections. What is more interesting is that cross-correlation is not monotonically increasing as a function of network layer; rather, it remains constant from layer 1 to all higher layers.

Figure 3 shows how rates and cross-correlations in higher layers vary as a function of input rates. In Figure 3A, we see that rates in higher layers are approximately linearly related to input rates, whereas in Figure 3B, we see that cross-correlations in higher layers depend on

input rates in an inverted-U shaped manner, with maximum cross-correlations occurring for input rates of 0.5.

We now briefly consider the case of spatially inhomogeneous inputs. As an example, let the mean rates of the second and third inputs into our feed-forward network,  $P_{20}$  and  $P_{30}$  or, in the notation of equation 3.5, p(2, 0) and p(3, 0), be equal, that is,  $p_{23} = P_{20} = P_{30}$ , where this equation defines  $p_{23}$ . Let us further assume that these inputs are correlated with a correlation coefficient  $q_{23}$ , with  $0 < q_{23} = 1$  (the case q = 0 is the case of spatially inhomogeneous, uncorrelated input, which is of less interest here). To simplify upcoming expressions, we make the following definitions:

$$q = \sqrt{q_{23}}$$
 (4.4)  
 $\bar{q} = (1 - q)$ . (4.5)

The equivalent of equation 4.3 for  $\overrightarrow{p}_{in}$  is now (see the appendix to Mikula and Niebur, 2003b, for detailed derivations)

$$\vec{P}_{in} = \begin{pmatrix} (1-P_{10})\left((1-p_{23})\left((1-p_{23})+p_{23}q\right)^{2}+p_{23}((1-p_{23})(1-q))^{2}\right)(1-P_{40})\\ (1-P_{10})\left((1-p_{23})\left((1-p_{23})+p_{23}q\right)^{2}+p_{23}((1-p_{23})(1-q))^{2}\right)P_{40}\\ (1-P_{10})2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})\bar{q}\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ (1-P_{10})2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ (1-P_{10})2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ (1-P_{10})\left((1-p_{23})\left(p_{23}\bar{q}q\right)^{2}+p_{23}(p_{23}+(1-p_{23})q)^{2}\right)(1-P_{40})\\ (1-P_{10})\left((1-p_{23})\left(p_{23}\bar{q}q\right)^{2}+p_{23}(p_{23}+(1-p_{23})q)^{2}\right)(1-P_{40})\\ P_{10}\left((1-p_{23})\left((1-p_{23})+p_{23}q\right)^{2}+p_{23}((1-p_{23})(1-q))^{2}\right)(1-P_{40})\\ P_{10}\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ P_{10}2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ P_{10}2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ P_{10}2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ P_{10}2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ P_{10}2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ P_{10}2\left((1-p_{23})p_{23}\bar{q}\left((1-p_{23})+p_{23}q\right)+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ P_{10}2\left((1-p_{23})\left(p_{23}\bar{q}q\right)^{2}+p_{23}\left(p_{23}+(1-p_{23})q\right)\left((1-p_{23})\bar{q}\right)\right)(1-P_{40})\\ P_{10}\left((1-p_{23})\left(p_{23}\bar{q}q\right)^{2}+p_{23}\left(p_{23}+(1-p_{23})q\right)^{2}\right)P_{40}$$

The results for the case of uncorrelated, spatially localized high input rates are shown in Figure 4. Figure 4A shows the mean rates for the input, even, and odd layers. Notably, spatially localized high input rates get propagated to arbitrarily high levels of the feedforward network. Besides the initial attenuation of firing rates at layer 1, there is no further attenuation. Figure 4B shows nearest-neighbor cross-correlations for the network, and Figure 4C shows cross-correlation matrices for each layer. Due to the convergence of

Neural Comput. Author manuscript; available in PMC 2012 November 13.

(4.6)

inputs to higher layers, uncorrelated, spatially localized high input rates appear as crosscorrelations in higher layers (see Figure 4B). That is, not only does rate information get propagated across the network, it also gets propagated as cross-correlation information.

The case shown in Figure 5 is identical to that in Figure 4 except that now the input rates are halved. Note that qualitatively, the results shown are identical to those of Figure 4, which means that for even relatively low spatially localized input rates, with p(2, 0) = p(3, 0) = 0.2, rate information still gets propagated to arbitrarily high levels of the feedforward network. In addition, this rate information also appears as cross-correlation information in higher layers.

Figure 6 shows the case for spatially localized cross-correlated inputs with uniform input rates. For this case, inputs 2 and 3 are cross-correlated to a value of  $q^2 = 0.4$ , and as can be seen in Figure 6B, this spatially localized cross-correlation gets propagated to arbitrarily high levels of the feedforward network. In Figure 6A, we see that the cross-correlation between inputs 2 and 3 is also represented in higher layers as higher rates. Note the appearance of increased cross-correlation between units 1 and 4 in layers 1 and higher in this and subsequent figures. It is due to the periodic boundary conditions and the specific connection scheme used: since input unit 2 projects to layer 1 unit 1, and input unit 3 projects to layer 1 unit 4, and units 2 and 3 are partially correlated, units 1 and 4 are also partially correlated but with a lower correlation coefficient than units 2 and 3, which both receive input from both correlated input units rather than from only one.

The case shown in Figure 7 is identical to that in Figure 6 except that now the crosscorrelation between inputs 2 and 3 is doubled to 0.8. This does not lead to much of an increase in higher-layer cross-correlations beyond what was seen in Figure 6B. Interestingly, in Figure 6A, we see that higher-layer rates are almost double what they were in Figure 6A. What this means is that doubling the input cross-correlation is reflected in an almost doubling of the higher-layer rates, as opposed to the relatively smaller increase in higherlayer cross-correlations.

Finally, in Figure 8, we see the case for spatially localized high cross-correlation (q = 0.6) and high firing rate between inputs 2 and 3. Figure 8A shows that rates are propagated with little attenuation, which is in contrast to the cases in Figures 4 and 5, where the high input rates were not cross-correlated. We see in Figure 8B that cross-correlations also get propagated across our network, though we also see relatively high cross-correlations between inputs 1 and 4 at higher layers, an effect likely attributed to the cyclic connectivity boundary conditions of our feedforward network.

#### **5 More Complex Networks**

The simple example of section 4 is useful for making explicit the details involved with the application of equations 3.4, 3.5, and 3.9. In this section, we look at a larger feedforward network, which has the advantage of being more realistic than the previous one, but with the trade-off that many of the details cannot be made explicit due to space limitations.

Our feedforward network is a slightly scaled-down version of the one introduced by Litvak, Sompolinsky, Segev, and Abeles (2003). We consider a 100-layer feedforward network containing exactly 1000 coincidence detectors per layer—500 excitatory and 500 inhibitory. The connectivity matrix is different between each layer<sup>3</sup> and is randomly constructed with

<sup>&</sup>lt;sup>3</sup>The connectivity used by Litvak et al. (2003) is apparently repeating (i.e., identical between each pair of subsequent layers). We studied this case as well and found results that were very similar to those with connectivity matrices that vary between each pair of layers and that are shown here.

Neural Comput. Author manuscript; available in PMC 2012 November 13.

precisely balanced excitation and inhibition such that each neuron receives exactly 50 excitatory inputs and 50 inhibitory inputs. Further, pairs of neurons in each layer share 5% of their inputs from the preceding layer. Each excitatory connection carries a weight of +1, and each inhibitory connection has a weight of -1.

We now define the inputs and the input states. For simplicity, we assume just two different input states. One state is defined with zero activity in all neurons of the input layer.<sup>4</sup> In the other state, 10% of the inputs provide perfectly correlated input spike trains (i.e., q = 1), with a spiking probability within each spike train of  $p_{in}$ . We designate these inputs as active inputs; since the total number of inputs is 1000, the number of active inputs is 100. All other input neurons (i.e., the other 90%) have zero spiking probability. Our requirement that the active inputs are perfectly correlated greatly simplifies what would otherwise be a difficult calculation. For example, if the active inputs were completely uncorrelated, then we would have  $2^{100}$  different input states to deal with, a clear impossibility. Assuming perfectly correlated active inputs is a simple and effective way to dramatically reduce the number of input states.

The results of this model using 100 different randomly generated connectivity matrices are shown in Figure 9 for average input spiking probabilities of .02 and .04 and using a threshold of 3. We show the results for only the first 20 layers of the network because higher layers are essentially the same as would be expected from Figure 9 (we studied up to layer 100). Spiking probabilities increase rapidly and immediately level off at about layer 2, remaining nearly constant for all higher layers. Because of the perfect correlation between all input neurons (q = 1), the mean firing rates at the higher layers of our network depend linearly on the initial input rates  $p_{in}$ . For example, in Figure 9a,  $p_{in} = .02$  results in a mean output spiking probability of .055 at higher layers, whereas in Figure 9b,  $p_{in}$  = .04 results in a mean output spiking probability of .11. Note the large standard deviations, indicating the richness and complexity of behavior for different connectivity matrices. The mean crosscorrelations are propagated in a similar manner as the mean rates shown in Figure 9: since all active neurons in our network are perfectly correlated with other, the mean crosscorrelation must be a function of the number of active neurons per layer, as is true for the mean rate per layer. Thus, if the fraction of active neurons in a given layer is r, the crosscorrelation in that layer is  $r^2$ . The propagating rates and cross-correlations show no sign of decrement at higher layers (layer 100 looks the same as layer 3 in terms of rates and crosscorrelations).

# 6 Discussion

This article extends our previous analytical results (Mikula & Niebur, 2003a, 2003b, 2004) for an individual coincidence detector to a feedforward network of coincidence detectors receiving steady-state cross-correlated binomial inputs at the zeroth layer. Thus, our derivation is valid only for steady-state neuronal responses and does not tell us anything about transient responses, which also appear to be important. For example, Diesmann, Gewaltig, and Aertsen (1999) studied temporal structures of spike trains in feedforward networks of spiking neurons. They showed that in different parameter regimes, synchronous packets of spikes can either travel from layer to layer without loss of coherency or, alternatively, disperse within a few layers. The objectives of their study are complementary to ours as it focuses on transient activity while ours is concerned with constant firing rates and correlations. Likewise, the approach taken by Diesmann et al. is complementary to what we have taken insofar as they use a more complex single-neuron model, a leaky integrate-and-fire neuron rather than the simpler coincidence detector, but it requires a numerical

<sup>&</sup>lt;sup>4</sup>For all nonzero thresholds, the same results would be achieved with all neurons active in the input layer.

solution while ours is analytical and recurrent and, furthermore, does not require introducing any approximations.

In spite of the obvious limitations of feedforward networks, they are of considerable interest for theoretical considerations. Likewise, coincidence detectors are an old and tested but simple model for the units of neural networks. However, although computational approaches allow the use of much more realistic neuron models (an extreme case perhaps being a recent study by Reyes, 2003, in which replicas of an actual biological neuron were used as the underlying units), the use of the much simpler coincidence detectors as building blocks allowed us to develop a closed form for the mean rate and pairwise cross-correlation of each neuron and neuron pair, respectively.

The details of our results apply only to the specific networks that we have studied here as examples to illustrate the general method we introduced. This method should, however, allow the analysis of other feedforward networks of coincidence detectors that are relevant for a specific problem. We emphasize that the methods described in section 3.3 permit one to find solutions for networks with much more realistic connectivity, including much higher numbers of synaptic connections.

One way in which our approach may prove useful is in the elucidation of the neural code, which involves addressing the question of how biological neural networks represent and transform information in their patterns of activity (Perkel & Bullock, 1968). Most research in the primate cortex has focused on two coding schemes, rate and temporal coding. In rate coding, information is coded purely in terms of average spiking activity, and variability of neural discharges is regarded as a form of noise. In temporal coding, neurons make use of the temporal structure of spike trains. Much experimental, computational, and theoretical work has been devoted to discussing this question, with evidence existing in favor of both rate codes and temporal codes.

One of the tools for studying these competing models is the numerical simulation of feedforward networks. Unfortunately, two such recent simulation results involving biologically inspired feedforward networks yielded conflicting results, in which the group conducting one study (Litvak et al., 2003) could not reproduce previously published numerical results of another group (Shadlen & Newsome, 1998). As we have shown in section 5, our approach can make a contribution to the resolution of this dispute, bearing in mind that the basic units employed in those studies are different from the coincidence detectors we employed. The availability of an analytical form for the solution provided by our approach clearly precludes similar disputes.

In a network of coincidence detectors, we find that aspects of both rate coding and temporal coding may be found. At least in the simple network studied in section 4, rate information is propagated up to arbitrarily high layers, even when spatially localized rate information is absent from the inputs and only spatially localized cross-correlation information is present. Also, in the case of uncorrelated inputs of uniform firing ratesshown in Figure 3A, we find that the output firing ratesof higher layers are almost linearly related to the input firing rates. Both would support that rate coding is at least a viable possibility, as proposed by Shadlen and Newsome (1998). The latter study may overemphasize the importance of rate coding, however: Salinas and Sejnowski (2000) pointed out that the influence of cross-correlations is relatively low in the Shadlen and Newsome study. The reason is that the latter study employed exactly equal correlation coefficients between inhibitory and excitatory populations ( $\rho_{EE} = \rho_{II} = \rho_{EI}$  in equation 20 of Salinas and Sejnowski, 2000), and the variance in the output layer becomes small since the contributions to the variance resulting from correlations between these populations cancel out ( $\rho_{EE} + \rho_{II} - 2\rho_{EI} = 0$ ; Salinas &

Sejnowski, 2000). Of course, the model described in section 5 works in the same parameter regime.

On the other hand, we find that rate coding is not the only possibility of propagating information across the layers of the network. Our results show that cross-correlation information is propagated across the feed-forward network to arbitrarily high layers, even when spatially localized cross-correlation information is absent from the inputs and only spatially localized rate information is present. Our results thus support the viability of temporal codes, as proposed by Litvak et al. (2003).

For the types of networks we studied in sections 4 and 5, we conclude that both rate and cross-correlation information is propagated across feedforward networks and, furthermore, that they will interact and tend to occur together, such that when only rate or only cross-correlation information is present in the inputs, this information will appear in higher layers in the form of both rate and cross-correlation information. As such, rate and cross-correlation codes may well be intertwined and interdependent.

An example of such a mixed code in complex nervous systems might be the representation of selective attention in the primate cortex (Niebur, 2002). Selective attention has been shown in electrophysiological studies to be correlated with both rate changes and changes in the fine temporal structure (on the order of milliseconds or tens of milliseconds) of neural activity (Moran & Desimone, 1985; Steinmetz et al., 2000; Fries, Reynolds, Rorie, & Desimone, 2001; Salinas & Sejnowski, 2001; Niebur, Hsiao, & Johnson, 2002). It will take more experimental as well as theoretical work to come to a conclusive answer as to which of the proposed neural coding schemes are used by the different nervous systems.

#### Acknowledgments

This work was supported by NIH/NINDS R01-NS43188-01A1 and NIH/NEI R01-EY16281.

#### References

- Diesmann M, Gewaltig MO, Aertsen A. Stable propagation of synchronous spiking in cortical neural networks. Nature. 1999; 402:529–533. [PubMed: 10591212]
- Fries P, Reynolds JH, Rorie AE, Desimone R. Modulation of oscillatory neuronal synchronization by selective visual attention. Science. 2001; 291:1560–11563. [PubMed: 11222864]
- Litvak V, Sompolinsky H, Segev I, Abeles M. On the transmission of rate code in long feedforward networks with excitatory-inhibitory balance. J. Neurosci. 2003; 23:3006–3015. [PubMed: 12684488]
- Mikula S, Niebur E. The effects of input rate and synchrony on a coincidence detector: Analytical solution. Neural Computation. 2003a; 15:539–547. [PubMed: 12625330]
- Mikula S, Niebur E. Synaptic depression leads to nonmonotonic frequency dependence in the coincidence detector. Neural Computation. 2003b; 15(10):2339–2358. [PubMed: 14511524]
- Mikula S, Niebur E. Correlated inhibitory and excitatory inputs to the coincidence detector: Analytical solution. IEEE Transactions on Neural Networks. 2004; 15(5):957–962. [PubMed: 15484872]
- Moran J, Desimone R. Selective attention gates visual processing in the extrastriate cortex. Science. 1985; 229:782–784. [PubMed: 4023713]
- Niebur E. Electrophysiological correlates of synchronous neural activity and attention: A short review. Biosystems. 2002; 67(1-3):157–166. [PubMed: 12459295]
- Niebur E, Hsiao SS, Johnson KO. Synchrony: A neuronal mechanism for attentional selection? Current Opinion in Neurobiology. 2002; 12(2):190–194. [PubMed: 12015236]
- Perkel D, Bullock TH. Neural coding. Neurosci. Res. Programm. Bull. 1968; 6:221.
- Reyes AD. Synchrony-dependent propagation of firing rate in iteratively constructed networks in vitro. Nat. Neurosci. 2003; 6:593–599. [PubMed: 12730700]

- Salinas E, Sejnowski TJ. Impact of correlated synaptic input on output firing rate and variability in simple neuronal models. J. Neurosci. 2000; 20(16):6193–6209. [PubMed: 10934269]
- Salinas E, Sejnowski TJ. Correlated neuronal activity and the flow of neural information. Nature Reviews Neuroscience. 2001; 2:539–550.
- Shadlen MN, Newsome WT. The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding. J. Neurosci. 1998; 18:3870–3896. [PubMed: 9570816]
- Steinmetz PN, Roy A, Fitzgerald P, Hsiao SS, Johnson KO, Niebur E. Attention modulates synchronized neuronal firing in primate somatosensory cortex. Nature. 2000; 404:187–190. [PubMed: 10724171]



#### Figure 1.

A simple *n*+1-layer feedforward network with cyclical boundary conditions, period 4. Coincidence detectors are represented by a circle with a  $\Theta(\Sigma)$  symbol in them, and connections are shown as directed arrows between coincidence detectors. Input is shown as stylized spike trains at the very bottom. Comma-separated pairs of numbers indicate layer (second number) and neuron in this layer (first number).



## Figure 2.

Effect of uniform, uncorrelated input firing rate. (A) Mean firing rates for the network. (B) Nearest-neighbor cross-correlations for the network. (C) Cross-correlation matrices for each layer.





(A) Higher-layer output rate versus input rate curve. (B) higher-layer cross-correlation versus input rate curve for the case of uncorrelated inputs of uniform rate.



#### Figure 4.

Effect of spatially localized high input rates. (A) Mean firing rates for the network. (B) nearest-neighbor cross-correlations for the network. (C) Cross-correlation matrices for each layer.



#### Figure 5.

Effect of halving input rates of Figure 4. (A) Mean firing rates for the network. (B) Nearestneighbor cross-correlations for the network. (C) Cross-correlation matrices for each layer.



#### Figure 6.

Effect of spatially localized high cross-correlation (q = .4) between inputs 2 and 3. (A) Mean firing rates for the network. (B) Nearest-neighbor cross-correlations for the network. (C) Cross-correlation matrices for each layer.



#### Figure 7.

Effect of spatially localized high cross-correlation (q = .8) between inputs 2 and 3. (A) Mean firing rates for the network. (B) Nearest-neighbor cross-correlations for the network. (C) Cross-correlation matrices for each layer.



#### Figure 8.

Effect of spatially localized high cross-correlation (q = .6) and high firing ratebetween inputs 2 and 3. (A) Mean firing rates for the network. (B) Nearest-neighbor cross-correlations for the network. (C) Cross-correlation matrices for each layer.

\$watermark-text



**Figure 9.** Mean layer output rates as a function of network layer. (A) Mean input rate of 0.2. (B) mean input rate of 0.4.

\$watermark-text

Neural Comput. Author manuscript; available in PMC	2012 November 1	3.

	ħ	3			<b>†</b> †	2			7 <b>4</b> 7	1			7 <b>4</b> ⊺	ر 0	
¥ 13	¥ ¥23	¥ ¥33	$\psi_{43}$	<b>¢</b> ↓ 12	$\psi_{22}$	$\psi_{32}$	$\psi_{42}$	¥ ¥11	$\psi_{21}$	$\psi_{31}$	$ec{m{\psi}}_{41}$	$ec{\psi}_{10}$	$\vec{\psi}_{20}$	¥ ¥30	$\psi_{40}$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	-	0	0	1	1	0	0	1	1	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0
-	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1
0	1	-	0	0	1	1	0	0	1	1	0	0	1	1	0
-	1	-	-	-	-	1	1	-	-	-	-	0	1		-
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
-	0	0	-	-	0	0	1	-	0	0	-	-	0	0	-
0	1	0	-	-	0	1	0	0	-	0	-		0		0
-	1	1	1	1	1	1	1	1	1	1	1	1	0	1	-
-	1	0	0	-	-	0	0	-	-	0	0	-	1	0	0
-	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-
-	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Mikula and Niebur

\$watermark-text