The network here is trained on learning a set of temporal sequences of outputs given a set of temporal sequences of inputs. The length of a sequence is 2000 timesteps. The network has 500 input units and 3 output units.

Training Set

The training set consists of 10 classes of input-output sequences – each class is based on a distinct set of input and target sequences which are perturbed slightly in order to generate different examples from that class. For a given class, we first randomly generate the canonical input & target output sequences which examples from that class will be based on. The canonical sequence for input unit i is given by a randomly generated sinusoidal curve:

$$x_i(t) = A_i \cos(B_i t + C_i) + 0.5 \tag{1}$$

where the amplitude factor A_i, the frequency B_i and the phase C_i are drawn from uniform distributions:

$$\begin{aligned} A_i &\sim \mathrm{U}(0.2, 0.4) \\ B_i &\sim \mathrm{U}(0.005, 0.03) \\ C_i &\sim \mathrm{U}(0, 1) \end{aligned} \tag{2}$$

The canonical target sequences for the 3 output units are generated in order to introduce some complexity in the functions that need to be learned – specifically, XOR functions. This is done because an XOR function cannot be properly learned without a hidden layer of neurons.

For each output unit j, we define the target sequence $\hat{y}_j^1$ as

$$\hat{y}_j^1(t) = \begin{cases} 0.8, & \text{if } x_a(t) > \gamma \text{ or } x_b(t) > \gamma \\ 0.2, & \text{if } x_a(t) > \gamma \text{ and } x_b(t) > \gamma \\ 0.2, & \text{if } x_a(t) \leq \gamma \text{ and } x_b(t) \leq \gamma \end{cases} \tag{3}$$

This is the XOR function applied to the thresholded versions of inputs $x_a(t)$ and $x_b(t)$, with a threshold γ, where a and b are different for each target unit. This creates a square wave, which is finally smoothed by fitting a cubic spline to $\hat{y}_j^1$.

a and b are chosen randomly for each output unit. The threshold γ is set to 0.5.

Thus, the target activity for each of the output units is to be active when either input x_a or x_b is active, but not when both are active, regardless of the activities of the other inputs (see Figure 1).

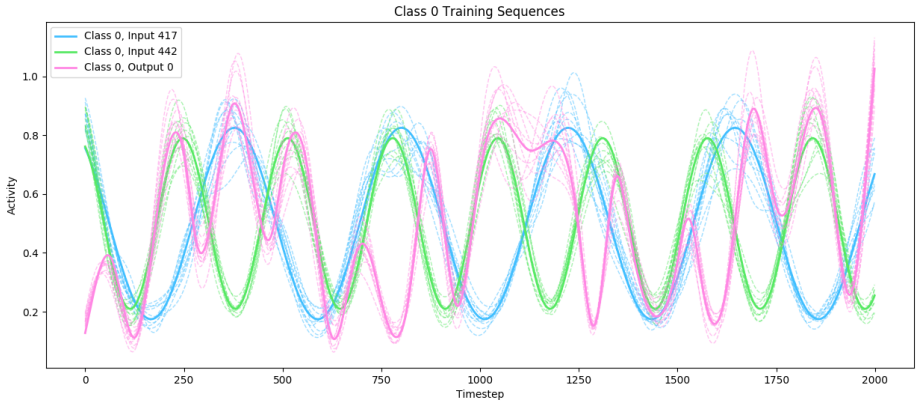

Figure 1: Activity sequence of output unit 0 for training class 0, and the activity sequences of the two input units which the target activity of output unit 0 was based on, $a = 417$ and $b = 442$. This shows the XOR function properties of the target activities – the target activity of output unit 0 is high when either input has high activity, but not when both do. *Solid lines:* Canonical sequences for the given class. *Dotted lines:* Training example sequences generated based on the canonical curves.

In order to generate training examples from this class, we add random variations to the amplitude and vertical shift of the canonical sequence curves. For input unit i we define:

$$\begin{aligned} a_i^x(t) &= A_i^x \cos(B_i^x t + C_i^x) + 1 \\ s_i^x(t) &= D_i^x \cos(E_i^x t + F_i^x) \end{aligned} \tag{4}$$

where

$$\begin{aligned} A_i^x &\sim \mathrm{U}(0.05, 0.2) \\ B_i^x &\sim \mathrm{U}(0.005, 0.05) \\ C_i^x &\sim \mathrm{U}(0, 1) \\ D_i^x &\sim \mathrm{U}(0.01, 0.05) \\ E_i^x &\sim \mathrm{U}(0.005, 0.05) \\ F_i^x &\sim \mathrm{U}(0, 1) \end{aligned} \tag{5}$$

Then, to generate an example sequence, $x_i(t)$ is adjusted so that:

$$x_i(t) \longrightarrow a_i^x(t) x_i(t) + s_i^x(t) \tag{6}$$

Similarly, for output unit j, we define:

$$\begin{aligned} a_j^{\hat{y}^1}(t) &= A_j^{\hat{y}^1} \cos(B_j^{\hat{y}^1} t + C_j^{\hat{y}^1}) + 1 \\ s_j^{\hat{y}^1}(t) &= D_j^{\hat{y}^1} \cos(E_j^{\hat{y}^1} t + F_j^{\hat{y}^1}) \end{aligned} \tag{7}$$

where

$$\begin{aligned} A_i^{\hat{y}^1} &\sim \mathrm{U}(0.05, 0.2) \\ B_i^{\hat{y}^1} &\sim \mathrm{U}(0.005, 0.05) \\ C_i^{\hat{y}^1} &\sim \mathrm{U}(0, 1) \\ D_i^{\hat{y}^1} &\sim \mathrm{U}(0.01, 0.05) \\ E_i^{\hat{y}^1} &\sim \mathrm{U}(0.005, 0.05) \\ F_i^{\hat{y}^1} &\sim \mathrm{U}(0, 1) \end{aligned} \tag{8}$$

and, to generate an example sequence, $\hat{y}_i^1(t)$ is adjusted so that:

$$\hat{y}_j^1(t) \longrightarrow a_j^{\hat{y}^1}(t) \hat{y}_j^1(t) + s_i^{\hat{y}^1}(t) \tag{9}$$

Figure 2 shows some of the input & target sequences generated for different classes, and training examples drawn from each class.

Network Structure and Dynamics

A diagram showing the network structure and dynamics is shown in Figure 3. Assume the network has l inputs, m hidden units and n output units. Unit j in the hidden layer has two compartments: a somatic compartment with voltage y_j^0 and an apical dendrite compartment with voltage g_j^0. At time t, $y_j^0(t)$ is given by:

$$y_j^0(t) = \sum_{k=1}^{l} W_{jk}^0 \tilde{x}_k(t-1) + b_j^0 \tag{10}$$

where $\boldsymbol{W}^0$ is the $m \times l$ matrix of the synaptic weights between the inputs and hidden layer units, $\boldsymbol{b}^0$ is a vector containing bias terms for each hidden unit, and $\tilde{\boldsymbol{x}}$ is the exponentially smoothed input layer activity:

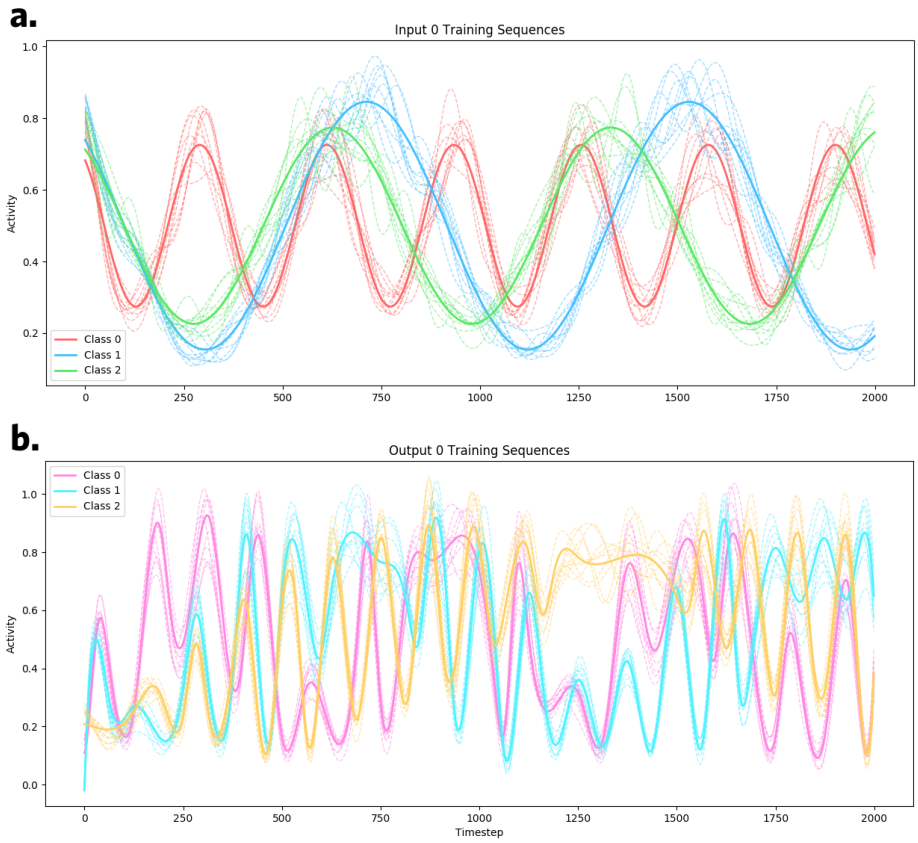

Figure 2: Input & target sequences for different classes, and sample sequences drawn from each class. **a.** Activity sequences of input unit 0 representing three different classes of training data. **b.** Activity sequences of output unit 0 representing three different classes of training data. *Solid lines:* Canonical sequences for the given class. *Dotted lines:* Training example sequences generated based on the canonical curves.

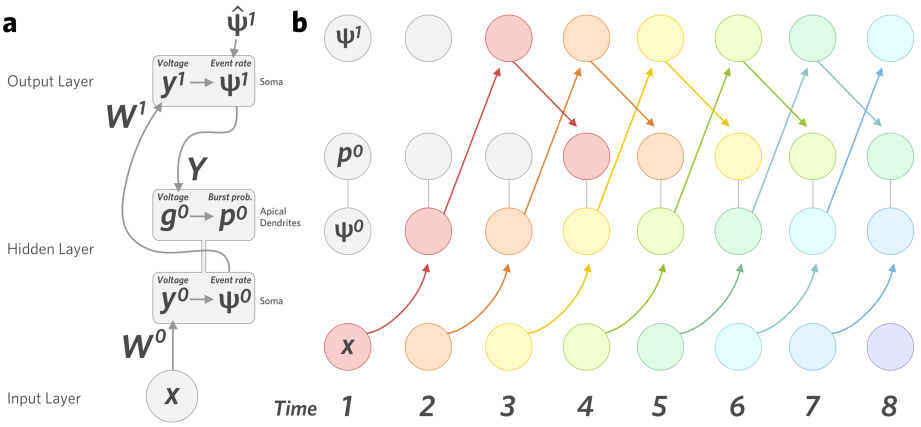

Figure 3: Diagram of the network. **a.** Network variables and connections. **b.** Temporal dynamics of the network.

$$\begin{aligned} \tilde{x}_k(0) &= x_k(0) \\ \tilde{x}_k(t) &= x_k(t) + \tilde{x}_k(t-1),\ t > 0 \end{aligned} \tag{11}$$

The hidden unit's *event rate* ψ_j^0, defined as the expected number of spike events (either single spikes or bursts) per unit time, is given by a sigmoid applied to the somatic voltage:

$$\psi_j^0(t) = \sigma(y_j^0(t)) = \frac{1}{1 + e^{-y_j^0(t)}} \tag{12}$$

This signal is received by units in the output layer. Unit i in the output layer has a somatic compartment with somatic voltage y_i^1 given by:

$$y_i^1(t) = \sum_{k=1}^{m} W_{ik}^1 \tilde{\psi}_k^0(t-1) + b_i^1 \tag{13}$$

where $\boldsymbol{W}^1$ are the feedforward synaptic weights between the hidden layer and output layer units, $\boldsymbol{b}^0$ are the bias terms for each output unit, and $\tilde{\boldsymbol{\psi}}^0$ are the exponentially smoothed event rates of the hidden layer units, computed as in equation (11). Similarly, the event rate of output unit i, ψ_i^1, is given by:

$$\psi_i^1(t) = \sigma(y_i^1(t)) = \frac{1}{1 + e^{-y_i^1(t)}} \tag{14}$$

Finally, the apical dendrite compartments of hidden layer units receive this signal from the output layer units. The apical voltage g_j^0 is given by:

$$g_j^0(t) = \sum_{k=1}^{n} Y_{jk} \tilde{\psi}_k^1(t-1) \tag{15}$$

where $\boldsymbol{Y}$ are the feedback synaptic weights between the output layer and hidden layer units, and $\tilde{\boldsymbol{\psi}}^1$ are the exponentially smoothed event rates of the output layer units, computed as in equation (11). The hidden unit's *burst probability* p_j^0, defined as the probability that a spike event will be a burst (rather than a single spike), is the given by applying the sigmoid function to the apical voltage:

$$p_j^0(t) = \sigma(g_j^0(t)) = \frac{1}{1 + e^{-g_j^0(t)}} \tag{16}$$

Training