The title

Carlo Alessi

May 24, 2019

Abstract

Supervised learning in networks of spiking neurons is efficiently carried out through a technique referred to as FORCE learning, at least for a certain range of frequency content in the teaching signals. As long as the teaching signals are limited to specific harmonics, FORCE can be efficiently used for learning and reproducing periodic signals [4], for chunking input streams [1], implementing signal routing [3], etc. The process however seems to break down when the teaching signals contain either low-frequency or very high-frequency components. This is due to the fact that the output of networks in the FORCE scheme is intimately bound to the intrinsic neuronal dynamics of these networks. This in turn currently restricts the applicability of the scheme to some toy examples, and prevents its more widespread use. In order to address these limitations, the following three questions are therefore of interest: 1) Can the FORCE scheme be adapted in order to learn sequences with low-frequency components, especially in the context of behaviourally-relevant motor tasks? 2) Can the FORCE scheme be extended in order to learn sequences and replay them at speeds that differ from the teaching signal? 3) Can the FORCE scheme be linked to Reinforcement Learning to derive an error/reward signal to drive the weight update process (which is more biologically plausible), instead of using explicit target functions (that are not biologically plausible)? One possible path to explore in this framework is to consider whether the requirement of the FORCE scheme to use a chaotic, non-saturated regime as a starting point for the learning process can be relaxed, and what that entails in terms of network structure, readout schemes, temporal constants of neuronal dynamics and learning processes. Some practical solutions can be implemented that go in that direction, and applied to motor control in the HBP Neurorobotics Platform (with a proposed emphasis on control of arm movements).

1 Background

1.1 Izhikevich model

The Izhikevich model is defined by a 2-D system of ordinary differential equations:

$$C\dot{v} = k(v - v_{rest})(v - v_{threshold}) - u + I \tag{1a}$$
$$\dot{u} = a(b(v - v_{rest}) - u) \tag{1b}$$

The variable $\boldsymbol{v}$ is the membrane potential. The variable $\boldsymbol{u}$ is the recovery variable. The parameter $\boldsymbol{C}$ is the membrane capacitance. The parameter $\boldsymbol{a}$ is the time scale of the recovery variable. The parameter $\boldsymbol{b}$ is the sensitivity of the recovery variable to the subthreshold fluctuations of the membrane potential. The resting membrane potential is $\boldsymbol{v_{rest}}$. The spiking threshold is $\boldsymbol{v_{threshold}}$. The parameter $\boldsymbol{k}$ controls the action potential up-swing. The synaptic current is $\boldsymbol{I}$.

If a spike occurs, the variables $\boldsymbol{v}$ and $\boldsymbol{u}$ are reset:

$$if\ v \geq v_{peak}\ then \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \tag{2}$$

The value $\boldsymbol{v_{peak}}$ is the voltage peak. The parameter $\boldsymbol{c}$ is the after-spike reset value of the membrane potential. The parameter $\boldsymbol{d}$ is the after-spike reset value of the recovery variable.

The discretized evolution of the system dynamics is given by:

$$v(t+1) = v(t) + \delta t \cdot [(k(v(t) - v_{rest})(v(t) - v_{threshold}) - u(t) + I)]/C \tag{3}$$
$$u(t+1) = u(t) + \delta t \cdot a[b(v(t) - v_{rest}) - u(t)] \tag{4}$$

The value $\boldsymbol{\delta t}$ is the integration time constant.

1.2 The FORCE method

The weight matrix used by the FORCE method [4] is decomposed by a static weight component $\boldsymbol{\omega^0}$ and a learned decoder $\boldsymbol{\phi}$:

$$\omega = G\omega^0 + Q\eta\phi^\intercal \tag{5}$$

The sparse and static weight matrix $\boldsymbol{\omega^0}$ induces chaos in the network. The chaos is balanced by the constant $\boldsymbol{G}$. The learned decoder $\boldsymbol{\phi}$ is determined online with Recursive Least Squares. The effects of the decoder is balanced by the constant $\boldsymbol{Q}$. The static encoding variable $\boldsymbol{\eta}$ defines the tuning preference of the neurons.

1.3 General network activity

A first-order differential equation implementing a low-pass filter is defined as:

$$\tau_a \dot{y} = x - y \tag{6}$$

The spike distribution $\boldsymbol{x(t)}$, which counts the number of spikes occurred at time t, is defined as:

$$x(t) = \sum_{i=1}^{N} \delta(t - t_i) \tag{7}$$

where

$$\delta(z) = \begin{cases} 1\ , & z = 0 \\ 0\ , & otherwise \end{cases} \tag{8}$$

The parameter $\boldsymbol{\tau_a}$ defines how much the spikes are smoothed by the filter, and should also impact the duration and/or frequency of the bursts.

By rearranging the terms in equation (6) it is possible to derive a discretized measure of the general network activity (GNA):

$$y(t+1) = \left(1 - \frac{\delta t}{\tau_a}\right) \cdot y(t) + \frac{\delta t}{\tau_a} \cdot x(t) \tag{9}$$

2 Methods

I edited the file IZFORCESINE.m under the directory "CODE FOR FIGURE 2/" in the repository `https://github.com/ModelDBRepository/190565`.

High-gain network The static weight matrix $\boldsymbol{\omega^0}$ was initialized from a Gaussian distribution with mean $\boldsymbol{\mu = 0}$ and standard deviation $\boldsymbol{\sigma = g/\sqrt{pN}}$, with connection probability $\boldsymbol{p = 0.1}$. The parameter $\boldsymbol{g}$ is the "gain" of the network and was set to a $\boldsymbol{g = 1.6}$ (high-gain) [2]. This value initializes the network to be on the edge of chaos.

2.1 Generating Oscillations of the GNA

Oscillations of the GNA variable y correspond to chuttering behaviour of the membrane potentials. There are three ways to generate oscillations: (i) by injecting an external sinusoidal wave (ii) using a global inhibition term, or (iii) by means of synaptic fatigue – a.k.a short term depression (STD).

External sinusoidal wave The easiest way to generate oscillatory behavior in the network activity is by injecting an external sinusoid wave of the form:

$$A \cdot \sin(\omega_{osc} \cdot t + \varphi_{osc}) \tag{10}$$

where A is the amplitude of the oscillations, ω_{osc} is the angular frequency, and φ_{osc} is the phase of the oscillations. Since we want to replicate the Hyppocampus, the sinuoid wave must resemble the so called theta oscillations, which frequency ranges between 4 and 10 Hz. In practice the oscillations are generated by adding at each time step t the value of (10) to the input current $I(t)$.

Global inhibition term The GNA variable $\boldsymbol{y}$ was used to implement a global inhibitoty current[1]. The dynamics of the membrane potential $\boldsymbol{v}$ in the Izhikevich model (1) were modified by adding the term $\boldsymbol{-\gamma y(t)}$, where $\boldsymbol{\gamma}$ is a tunable parameter, obtaining:

$$C\dot{v} = k(v - v_{rest})(v - v_{threshold}) - u + I - \gamma y \tag{11}$$

$$\dot{u} = a(b(v - v_{rest}) - u) \tag{12}$$

3 Experiments

3.1 Pre-training

Figure 1, Figure 2

3.2 Training

We refer to *continuous training* when the weight update is performed every Δt milliseconds. We refer to *phase training* when the weights are updated at a specific point of the phase of the target signal, only once per cycle.

3.2.1 Changing the phase φ_{sin} of the teaching signal

Determine whether the phase of the teaching signal $\varphi_{sin} \in \{0\pi, \frac{1}{2}\pi\}$ has an impact on the performance. The network is first stabilized for 5 seconds of simulation. The network is then trained for an amount of time corresponding to 20 cycles of the target signal. This choice was made to better compare the performance obtained with target signals of different frequencies $f \in \{0.3, 0.5, 0.8, 1, 5, 7, 12\}$ Hz. The frequency of the external oscillations was fixed at $f = 4$ Hz, whereas it was varied the amplitude $A \in \{200, 500\}$.

When FORCE training was turned off, the simulation was carried out for additional 10 target signal cycles. For each setting, it is computed the Average Firing Rate and the Average Squared Error between the target signal and the approximant, over the 10 post training cycles. Each result is obtained as the average of 10 repetitions. The Results are summarized in Table 1 and Table 2.

Discussion Our approach performs slightly better than the Clopath settings for low frequency target signals, vice versa for high frequency target signals.

References

[1] Toshitake Asabuki, Naoki Hiratani, and Tomoki Fukai. Interactive reservoir computing for chunking information streams. *PLoS computational biology*, 14(10):e1006400, 2018.

[2] Vishwa Goudar and Dean V Buonomano. Encoding sensory and motor patterns as time-invariant trajectories in recurrent neural networks. *Elife*, 7:e31134, 2018.

[3] Gregor M Hoerzer, Robert Legenstein, and Wolfgang Maass. Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. *Cerebral cortex*, 24(3):677–690, 2012.

[4] Wilten Nicola and Claudia Clopath. Supervised learning in spiking neural networks with force training. *Nature communications*, 8(1):2208, 2017.

[1] Same for all neurons.

Table 1: Comparison of Average Firing Rate and Average Error obtained by Our's and Clopath's. The external sinuoid amplitude was $A = 200$. The teaching signal phase was $\varphi_{sin} \in \{0\pi, \frac{1}{2}\pi\}$. Training lasted for 20 cycles of the teaching signal. The results were averaged over 10 cycles after training, for 10 trials, plus/minus standard deviation shown. Best results shown in bold.

A=200

$\boldsymbol{\varphi_{sin} = 0\pi}$	Average Firing Rate		Average Error	
f_{sin}	Our	Clopath	Our	Clopath
0.3	24.6090 +/- 1.1663	28.9982 +/- 2.0957	**0.8720 +/- 0.1207**	0.9719 +/- 0.1663
0.5	28.0870 +/- 1.1094	29.7849 +/- 0.6220	1.0613 +/- 0.1370	**0.9941 +/- 0.2052**
0.8	27.4801 +/- 0.9859	28.2724 +/- 0.8951	0.9791 +/- 0.2498	**0.8373 +/- 0.4707**
1	27.6185 +/- 0.9346	28.8081 +/- 0.9722	**0.2633 +/- 0.3529**	0.6576 +/- 0.5382
5	27.3886 +/- 1.4616	36.5213 +/- 0.8475	0.4951 +/- 0.3416	**0.0483 +/- 0.0306**
7	29.9600 +/- 1.8270	36.6402 +/- 0.9950	0.4806 +/- 0.2318	**0.0547 +/- 0.0370**
12	25.4990 +/- 3.0251	37.3254 +/- 0.6670	0.3226 +/- 0.1534	**0.0171 +/- 0.0192**
$\boldsymbol{\varphi_{sin} = \frac{1}{2}\pi}$	Average Firing Rate		Average Error	
f_{sin}	Our	Clopath	Our	Clopath
0.3	24.3086 +/- 0.8388	28.6349 +/- 1.4053	**0.8643 +/- 0.1159**	0.9425 +/- 0.1224
0.5	27.4820 +/- 1.2971	29.0829 +/- 1.0393	1.0215 +/- 0.1343	**0.9310 +/- 0.1149**
0.8	26.6556 +/- 1.3026	28.0101 +/- 1.0852	**0.9922 +/- 0.1482**	1.0324 +/- 0.2474
1	27.8157 +/- 0.6580	28.3632 +/- 0.8934	**0.2773 +/- 0.3594**	0.5254 +/- 0.4243
5	28.6744 +/- 0.8461	37.1198 +/- 0.7418	0.6558 +/- 0.3874	**0.0229 +/- 0.0296**
7	30.6412 +/- 2.3428	37.6164 +/- 0.7145	0.6007 +/- 0.3550	**0.0198 +/- 0.0171**
12	25.8433 +/- 2.0667	37.8090 +/- 0.8193	0.3793 +/- 0.1573	**0.0053 +/- 0.0047**

Table 2: Comparison of Average Firing Rate and Average Error obtained by Our's and Clopath's. The external sinuoid amplitude was $A = 500$. The teaching signal phase was $\varphi_{sin} \in \{0\pi, \frac{1}{2}\pi\}$. Training lasted for 20 cycles of the teaching signal. The results were averaged over 10 cycles after training, for 10 trials, plus/minus standard deviation shown. Best results shown in bold.

A=500

$\boldsymbol{\varphi_{sin} = 0\pi}$	Average Firing Rate		Average Error	
f_{sin}	Our	Clopath	Our	Clopath
0.3	26.5297 +/- 0.6659	28.9982 +/- 2.0957	**0.9114 +/- 0.1023**	0.9719 +/- 0.1663
0.5	28.2868 +/- 0.8022	29.7849 +/- 0.6220	**0.8308 +/- 0.0999**	0.9941 +/- 0.2052
0.8	27.8620 +/- 0.8376	28.2724 +/- 0.8951	**0.8218 +/- 0.2159**	0.8373 +/- 0.4707
1	28.7076 +/- 0.6874	28.8081 +/- 0.9722	**0.3138 +/- 0.3756**	0.6576 +/- 0.5382
5	26.9306 +/- 1.9153	36.6349 +/- 0.7633	0.7100 +/- 0.2743	**0.0403 +/- 0.0265**
7	26.9202 +/- 1.1519	36.6402 +/- 0.9950	0.6365 +/- 0.1424	**0.0547 +/- 0.0370**
12	21.4647 +/- 1.4200	37.3254 +/- 0.6670	0.2852 +/- 0.0582	**0.0171 +/- 0.0192**
$\boldsymbol{\varphi_{sin} = \frac{1}{2}\pi}$	Average Firing Rate		Average Error	
f_{sin}	Our	Clopath	Our	Clopath
0.3	26.4543 +/- 0.9647	28.6349 +/- 1.4053	**0.9063 +/- 0.1121**	0.9425 +/- 0.1224
0.5	28.1135 +/- 0.7884	29.0829 +/- 1.0393	**0.9030 +/- 0.1271**	0.9310 +/- 0.1149
0.8	28.7706 +/- 1.1242	28.0101 +/- 1.0852	**0.9326 +/- 0.3808**	1.0324 +/- 0.2474
1	29.1753 +/- 0.6522	28.3632 +/- 0.8934	0.6804 +/- 0.6225	**0.5254 +/- 0.4243**
5	26.2176 +/- 1.3869	37.1198 +/- 0.7418	0.7559 +/- 0.2753	**0.0229 +/- 0.0296**
7	26.3405 +/- 2.1442	37.6164 +/- 0.7145	0.6352 +/- 0.1746	**0.0198 +/- 0.0171**
12	22.6126 +/- 0.8991	37.8090 +/- 0.8193	0.2550 +/- 0.0136	**0.0053 +/- 0.0047**

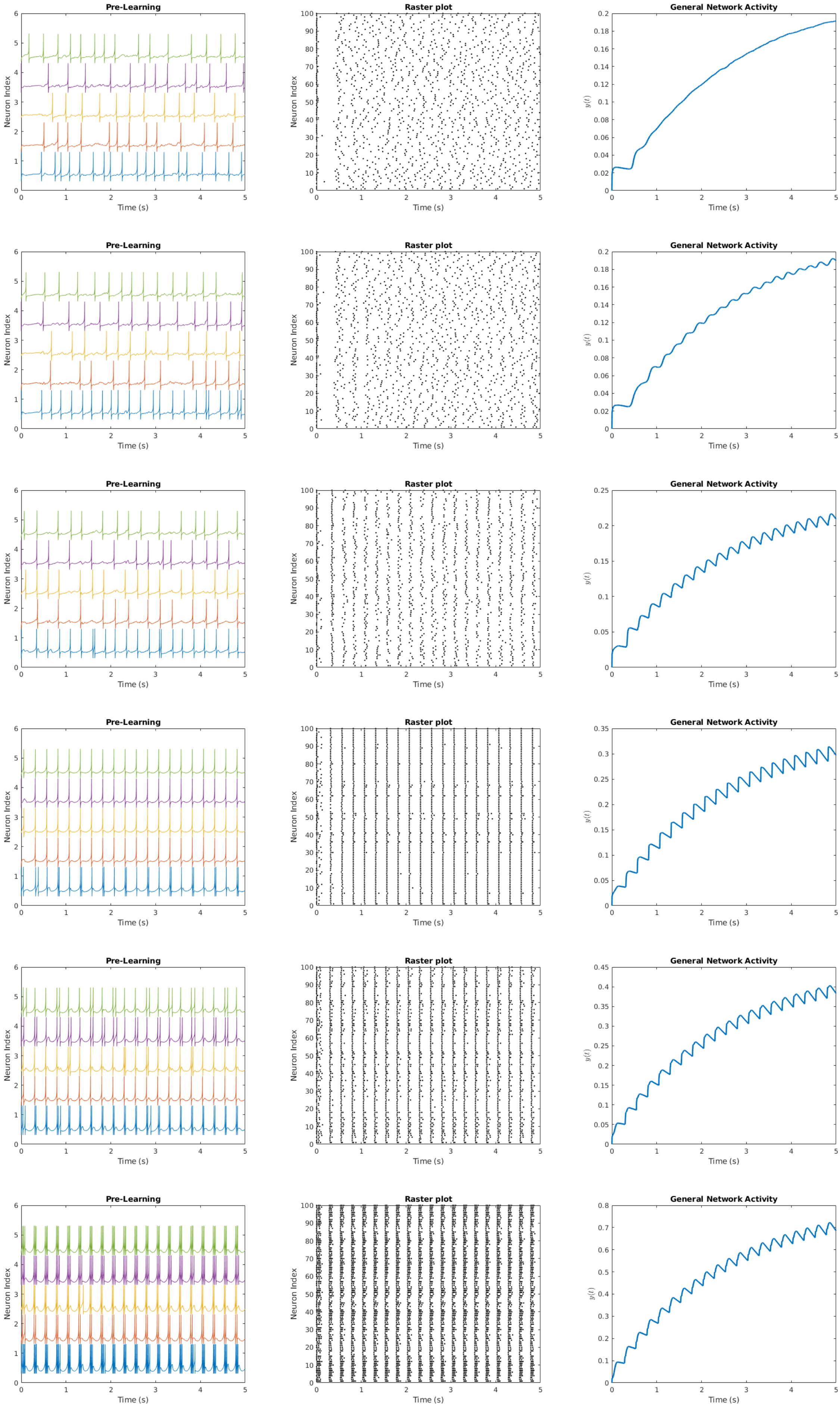

Figure 1: Pre-learning activity chaging amplitude of external oscillations, $A \in \{0, 10, 50, 100, 200, 500\}$.

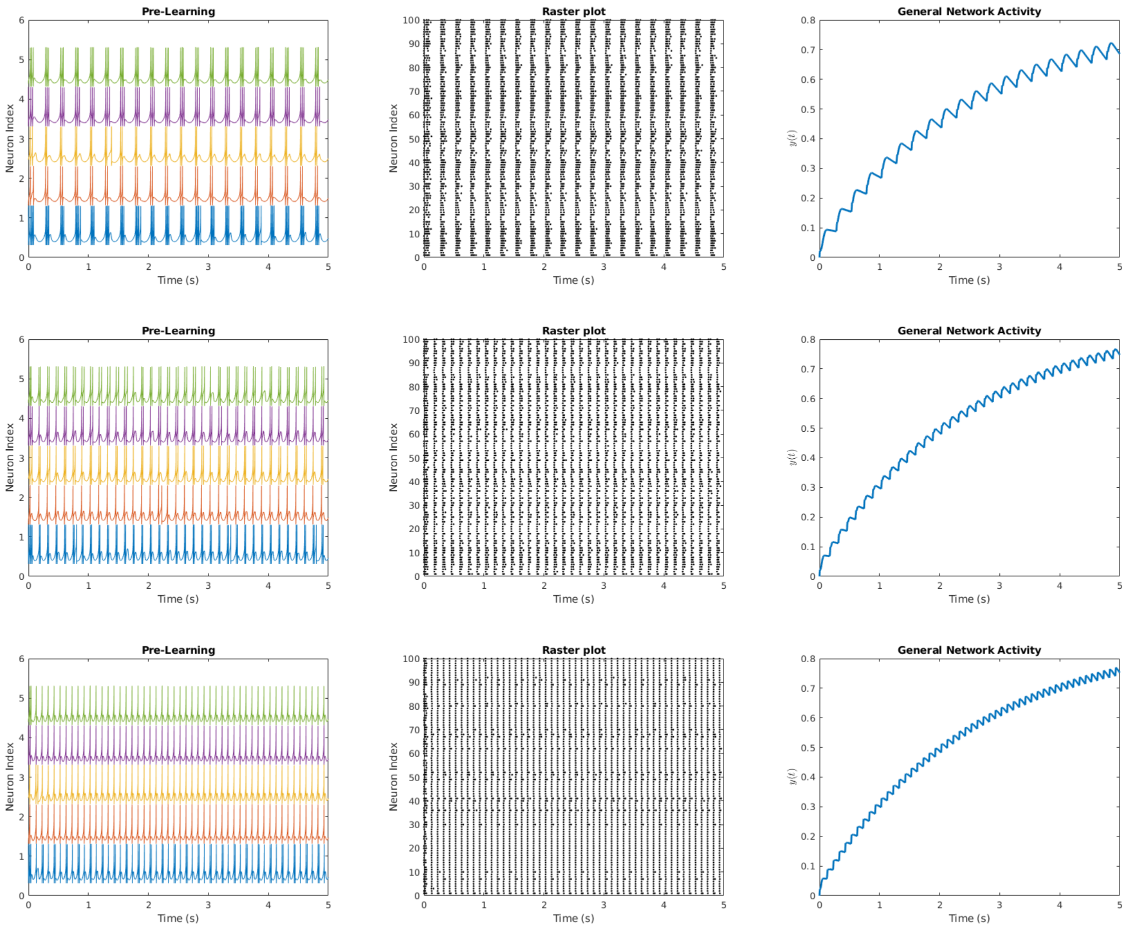

Figure 2: Pre-learning activity changing frequency of external oscillations $f \in \{4, 7, 10\}$ Hz.