Notes: Variability and oscillations

M. Rule

January 7, 2020

These notes outline a stochastic binary neural network as a toy model of the effects of noise on neural computation. We explore how noise modifies the effective gain in linear-nonlinear networks, and how oscillations might modulate some of the effects of noise.

Spiking responses in biological neural networks are variable: repeated presentations of the same stimulus can trigger different activity patterns. Some variability arises from the biological nature of neurons, and some may relate to ongoing processing. Does the brain use this variability for computation? What processes are available to modulate or attenuate variability? In these notes, we consider a mathematically convenient model of variability in spiking neural networks. This allows us to explore how noise propagates and affects computation. We explore two main phenomenon: the modulation of neuronal properties by noise ("stochastic gain modulation"), and a role for network oscillations in attenuating the impact of noise on neural computation.

0.0.1 1. Stochastic binary networks and the dichotomized Gaussian

In this section, we describe both deterministic and stochastic feed-forward binary neural networks, which will serve as a model of neural computation.

Consider linear-nonlinear "neurons" ("units"), which take a vector of inputs x. These inputs are multiplied by a weights w, summed up, and passed through a nonlinearity f (sometimes called the "activation function" or "transfer function"), to yield some output.

We first consider a deterministic binary unit, which uses the Heaviside step function $H(x)$ for the nonlinearity (Fig. 1a). The binary output $s \in \{0, 1\}$ is given by the expression $s = H(w^\top x + b)$, where b is a bias parameter which sets the threshold. Multi-layer binary networks can compute Boolean functions and solve basic classification tasks (Fig. 1b). Such networks are a minimalist model of spike-based neural computation, since the spikes that neurons use to communicate are all-or-nothing events.

Neurons in real biological neurons are affected by a multitude of unobserved processes that make spiking stochastic (Faisal et al. 2008). To model this in a binary network, we make the spiking a Bernoulli variable conditioned on some latent rate p (Fig. 1c):

$$\begin{aligned} a &= w^\top x + b \\ p &= f(a) \\ s &\sim \text{Bernoulli}(p), \end{aligned} \tag{1}$$

where f is some nonlinearity with output $p \in [0, 1]$ between zero and one. We've also explicitly defined a as the net synaptic "activation" of the nonlinear unit, since treating this variable separately will be useful in later derivations. Often, f is taken to be the logistic function $f(a) = 1/(1 + e^{-a})$.

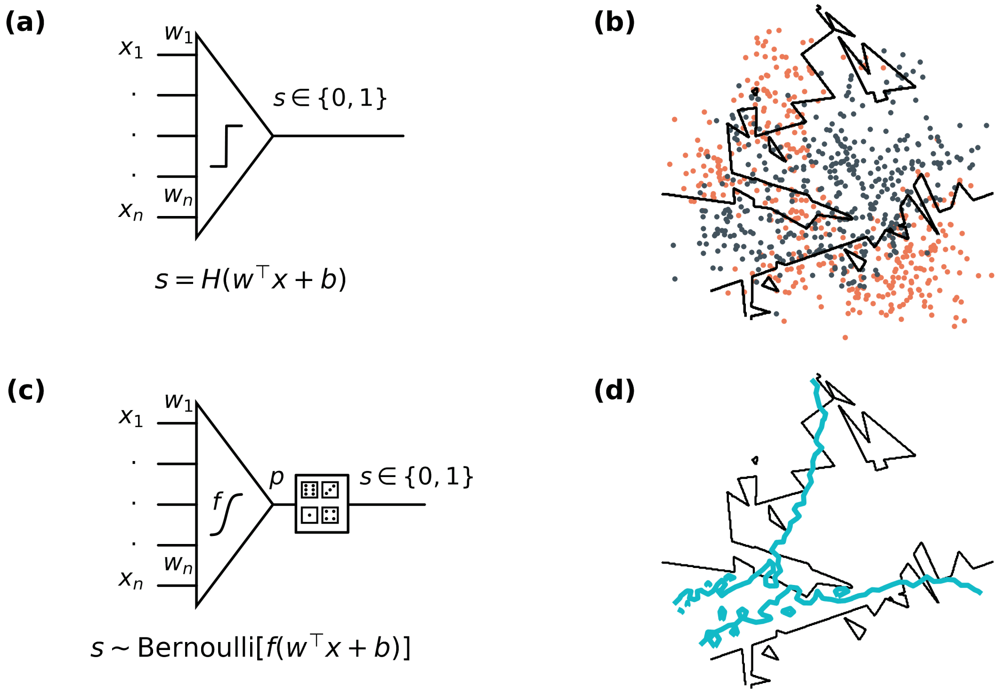

Figure 1: Binary neural networks with and without noise (a) A binary threshold unit accepts inputs x, computes a weighted sum of these inputs $w^\top x$, and emits a "1" if this sum exceeds an internal threshold b, and "0" otherwise **(b)** Networks of these units can perform simple binary classification tasks. In this toy model, we initialized 50 random binary-threshold features and identified a hard classification boundary via regression on these features. Points indicate training data, colored according to class label. **(c)** We can introduce noise by making the binary output a stochastic Bernoulli variable, in which each unit computes a "spiking probability" p, and emits a "1" with probability p, and a "0" with probability $1 - p$. **(d)** Stochasticity in spiking makes the classifier probabilistic. It also distorts the classification boundaries. Shown here is the original classification boundary (black), and an estimate of the 50% classification threshold from 5K samples from a stochastic network (blue).

This model amounts to a linear-nonlinear-Bernoulli model, and can be viewed as a discrete-time analogue of the linear-nonlinear-Poisson models used in point-process modeling of spike trains (Truccolo et al. 2005, Pillow et al. 2008, Truccolo 2010, 2016, Ostojic and Brunel 2011).

Noise in a binary network can can also disrupt computational properties, like the location of classification boundaries (Fig. 1d). This is due to the interaction of noise with firing rate variability.

For example, consider a firing-rate nonlinearity with positive curvature, say $f(a) = \exp(a)$. Nonlinearities with positive curvature amplify positive fluctuations in a more than negative ones. The presence of variability in a therefore increases the mean rate of the neuron, disrupting the function being computed.

A convenient model for stochastic binary neurons arises from considering a deterministic threshold unit with some Gaussian noise affecting its spiking threshold (Fig. 2a). This is the "dichotomized Gaussian" (DG) model (Pearson 1909, Emrich and Piedmonte 1991, Cox and Wermuth 2002), and has proven an elegant model for variability and correlations in spiking networks (Matthias and Berens 2008, Macke et al. 2009, 2011). In the dichotomized Gaussian model, variability in the spiking output arises from variability in the effective threshold of the neuron (Fig 2b).

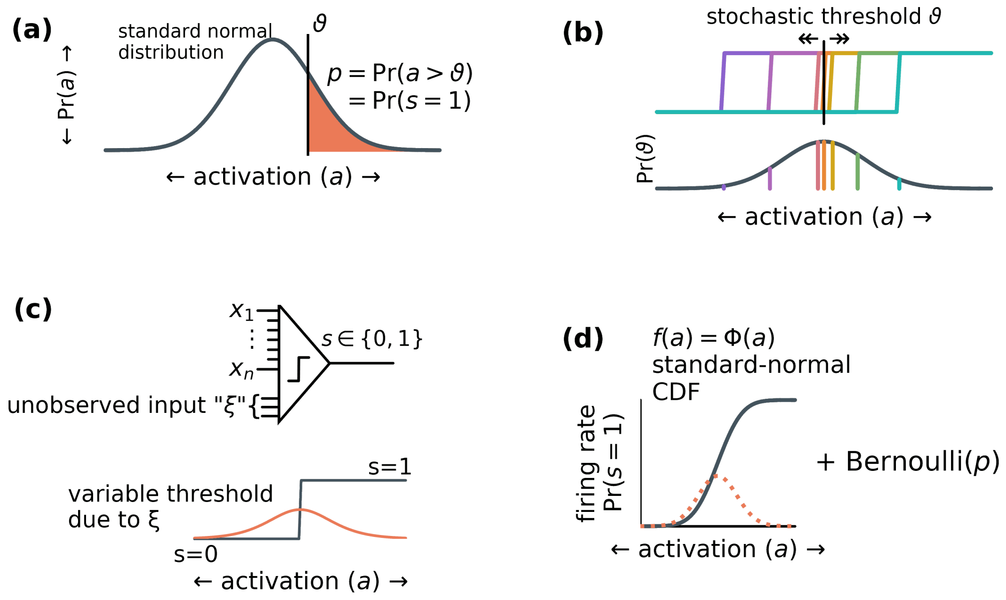

Figure 2: The dichotomized Gaussian model for stochastic binary neurons (a) One can sample a Bernoulli random variably by drawing a Gaussian random variable, and emitting "1" if it exceeds a threshold ϑ, and "0" otherwise. **(b)** This is analogous to considering a deterministic (hard-threshold) binary unit, with some additional Gaussian noise affecting the threshold value **(c)** Stochasticity in binary units can also be viewed as a result of unobserved inputs in an otherwise deterministic threshold network. The unobserved inputs can make spiking appear to be stochastic, if only a limited number of inputs are observed. **(d)** One may model this as a Bernoulli neuron, where the firing probability p is computed by first taking a weighted sum of the inputs, applying some bias, and then passing it through a saturating nonlinearity $\Phi(w^\top x + b)$, which is the cumulative distribution function of a standard Gaussian distribution.

So far, we have described stochasticity as an effect on the output, as if each unit flips a (biased) coin to decide whether to spike (emit a "1") or not (emit a "0"). However, isolated neurons cab respond reliably to injected current. Variability in biological neural networks may reflect propagation of variability in the inputs, the influence of unobserved inputs, and the result of stochastic synaptic transmission. One may therefore also view stochastic binary "neurons" as deterministic threshold units, which receive some unobserved, uncorrelated noise, leading to a stochastic threshold (Fig. 2c).

For the purposes of these notes, we model the noise ξ as coming from a standard normal distribution:

$$\begin{aligned} a &= w^\top x + b \\ s &= H(a + \xi) \\ \xi &\sim \mathcal{N}(0, 1) \end{aligned} \tag{2}$$

Equivalently, one may interpret this as a Bernoulli neuron that uses the standard normal cumulative distribution function (CDF), $\Phi(a)$, as its firing rate nonlinearity (Fig. 2d). This is analogous to a "Probit" regression model:

$$\begin{aligned} a &= w^\top x + b \\ p &= \Phi(a) \\ s &\sim \text{Bernoulli}(p) \\ \Phi(a) &= \int_{-\infty}^{a} (2\pi)^{-\frac{1}{2}} e^{-\frac{u^2}{2}} du. \end{aligned} \tag{3}$$

The standard-normal CDF is also a sigmoidal nonlinearity, and sufficiently similar to the logistic function that they can be treated as approximations of each-other (with appropriate scaling of inputs). For simplicity, we consider neurons with a Gaussian-CDF nonlinearity directly, rather than treating this model as an approximation for neurons with a logistic nonlinearity.

0.0.2 2. Computing in stochastic binary networks

In this section, we explore the dichotomized Gaussian model of stochastic binary neural networks, which lets us predict how noise affects computation. We also explore strategies for making computation robust to noise.

2.1 *Noise decreases gain in stochastic binary neurons* Noise in the input to a nonlinear neuron can cause a shift in the mean output (Fig. 3a). The influence of noise in a stochastic binary unit can be likened to a decrease in the effective gain (steepness) of the nonlinear transfer function (Fig. 3b). For illustration, consider a scenario in which the neuronal inputs (and therefore neuronal activation $a \sim \mathcal{N}(\mu_a, \sigma_a^2)$) are themselves stochastic. This is equivalent to a scenario where the activation is deterministic, $a = \mu_a$, but corrupted by additional noise v:

$$\begin{aligned} p &= \Phi(\mu_a + v) \\ v &\sim \mathcal{N}(0, \sigma_a^2). \end{aligned} \tag{4}$$

Intuitively, this additional variability "spreads out" the neuronal activation, which is equivalent to decreasing the steepness (gain) of the transfer function.

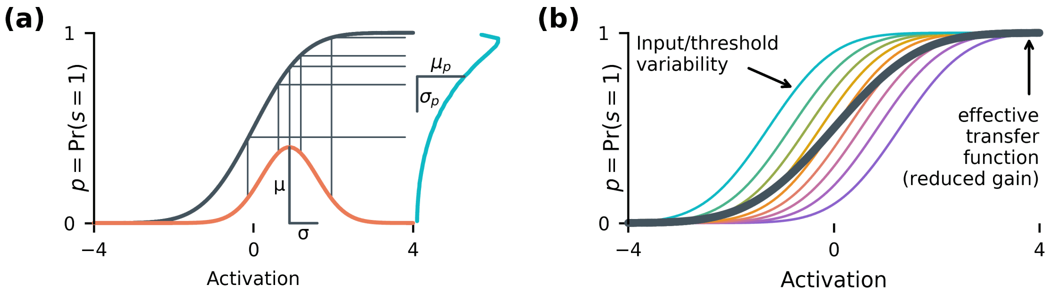

Figure 3: Noise propagation in stochastic neurons with sigmoidal nonlinearity (a) One can predict how much variability and noise is present in the output, given the covariance structure of the inputs. This allows us to reason about how variability propagates to

downstream neurons, and how it affects neural computation. **(b)** The influence of noise (or input variability) can be modeled as a decrease in gain of the nonlinear transfer function. Analytic expressions for this change in gain are possible in the case of Gaussian noise and a dichotomized-Gaussian neuron.

To see this analytically, consider the dichotomized Gaussian formulation of a stochastic neuron. Noise sources include the intrinsic threshold noise $\xi\sim\mathcal{N}(0,1)$, as well as added noise arising from inputs $v\sim\mathcal{N}(0,\sigma^2)$, and so the total noise has variance $1+\sigma^2$. The Heaviside step function doesn't change if we rescale its input, so we can change variables and write:

$$\begin{aligned} s &= H(a+\xi+v) \\ &= H\left(\gamma a+\xi\right) \\ \gamma &= 1/\sqrt{1+\sigma^2} \\ \Rightarrow & \\ p &= \Phi(\gamma a) \\ s &\sim \text{Bernoulli}(p), \end{aligned} \tag{5}$$

where the gain parameter γ modulates the steepness of the sigmoidal nonlinearity. That is, the influence of additional noise can be treated as a gain modulation (Fig. 3b). This amounts to a nonlinear Bernoulli neuron where the firing-rate nonlinearity is given by p=$\Phi(\gamma a)$, and provides an expression for the first moment (mean) of the output, $\langle s\rangle = p$, in terms of the moments of the activation μ, σ^2.

When the gain γ is large, p is almost always very close to 0 or very close to 1. In the limit where $\gamma \longrightarrow \infty$, we recover a deterministic binary neuron with a Heaviside step nonlinearity, $\lim_{\gamma\to\infty}\Phi(\gamma a) = H(a)$. In the limit $\gamma \longrightarrow 0$, spiking becomes completely stochastic ($p = 0.5$) and is unrelated to the input.

The high-gain limit highlights a useful phenomenon in stochastic binary neural networks: if neurons are either strongly suppressed ($p{\approx}0$), or driven to saturation ($p{\approx}1$), then very little spiking noise is added, since spiking becomes almost deterministic. Patterns of silence can therefore be as important as robust spiking, for precise reliable neural coding (Schneidman et al. 2011). In section 3, we will use this property to show how oscillatory drive might attenuate threshold noise.

There are physiological limits to how much one can increase gain (and therefore decrease noise). The neuronal integration and spiking mechanisms are subject to thermal fluctuations, placing a lower limit on the noise. We interpret ξ as a minimum noise floor, and choose units such that ξ has unit variance. Noise also arises from unreliable vesicle release in presynaptic terminals, which cannot be controlled by the postsynaptic neuron, and noise propagated from the inputs themselves cannot be attenuated independently of the signal.

2.2 *Training binary neural networks with noise* Eliminating noise is challenging, and presumably metabolically expensive. Could networks to learn to produce a target output without attenuating internal fluctuations? Could noise itself be computationally useful? These questions have been reviewed in depth elsewhere, but we briefly revisit them here.

As we saw in the dichotomized Gaussian model, noise can transform a hard threshold into a soft, probabilistic one, which allows for graded computations with binary units (Fig. 2). If hard nonlinearities are easier or cheaper to build, noise could therefore provide an efficient approach to constructing a smooth, sigmoidal nonlinearity. This "softening" of the nonlinearity also allows one to train artificial neural networks that would not otherwise be amenable to conventional approaches.

Noise may also be useful in neural networks that perform statistical sampling. In this scenario, the output of the network is inherently variable, and this variability reflects statistical uncertainty about the encoded quantities (e.g. Echeveste et al. 2019).

In binary neural networks, the slope of the effective nonlinearity depends on the magnitude of the input noise (Fig. 3). Such noise-mediated gain modulation could be computationally useful, as it allows another mode of nonlinear interaction between neurons.

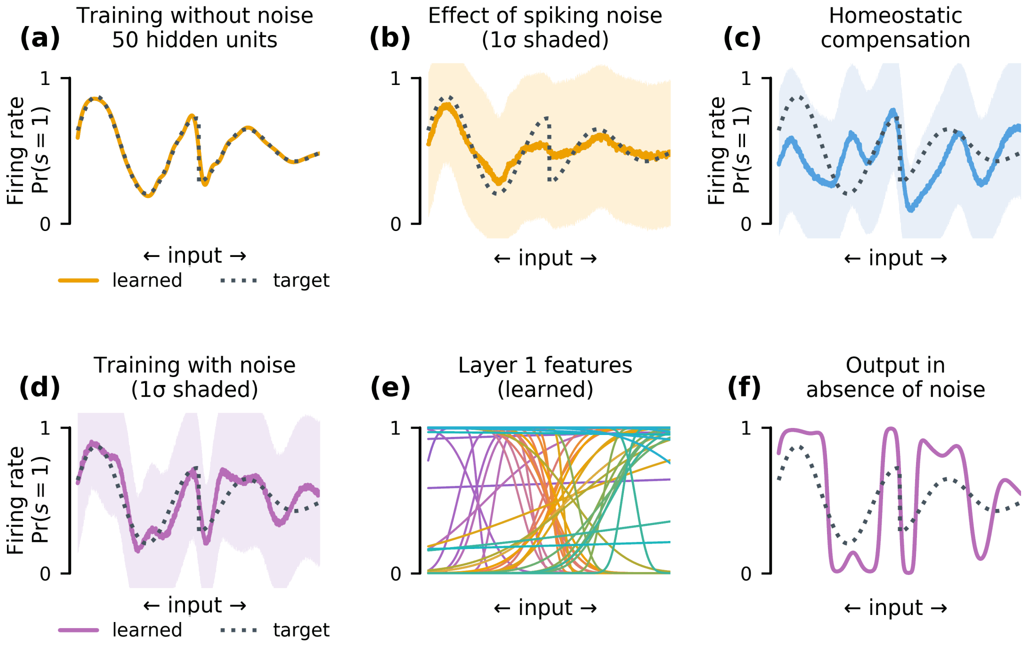

Figure 4: Re-training a neural network for noise (a) We trained a deterministic network to reproduce a target scalar function of one variable. Units were deterministic with a sigmoidal nonlinearity. There were 50 hidden units and a single readout neuron. **(b)** The deterministic network fails if spiking noise is added. The mean output of the trained output is shown in yellow, and the shaded region represents the trained output variability $\pm 1\sigma$, estimated via Monte-Carlo sampling. **(c)** Adjusting the gain and bias of the readout neuron to restore the statistics of spiking input rescues the original computation. **(d)** Learning can also compensate for the effects of noise on computation. Here, we trained a linear-nonlinear-Bernoulli (LNB) network by maximizing the likelihood of the data under a dichotomized-Gaussian moment approximation. **(e)** The moment representation allowed backpropagation of errors to optimize the input features of the network (20 features, mean activations plotted here). **(f)** If the noise is removed from the re-trained network, performance again degrades. Correct operation of the circuit now requires noise.

In Figure 4, we illustrate the importance of modeling noise when training stochastic binary networks. The output of a deterministic network with a sigmoidal nonlinearity (Fig. 4a) breaks down when spiking noise is added (Fig. 4b).

For a single neuron, this disruption is well-modeled by a noise-induced change in gain. It is therefore possible to partially rescue the original computation by homeostatically adjusting the synaptic gains to compensate for noise (Fig. 4c).

If the same network is trained in the presence of noise, the correct mean output occurs despite intrinsic spiking fluctuations (Fig. 4c). To train in the presence of noise, we used form of backpropagation based on moment approximations of dichotomized Gaussian neurons. This allowed the network to learn input-layer features based on the output error (Fig. 4e).

Computations learned in the presence of nose are sensitive to changes in the statistics of said noise. If the threshold noise is removed from our stochastic neural network, the computed function is altered (Fig. 4f). If threshold noise is used for computation, one might therefore expect regulatory mechanism to stabilize the amount of noise in the network.

0.0.3 3. Could oscillations moderate neuronal variability?

So far, we have demonstrated that the impact of noise on computation can be viewed as a noise-related disruption in the transfer functions of single neurons. If the statistics of noise are predictable, networks can learn to be robust to noise. Homeostatic mechanisms might confer additional robustness, if the rate of change in noise statistics is gradual. Is it possible to cancel unexpected fluctuations in noise levels on more rapid timescales?

One of the reasons homeostatic mechanisms are slow (and can only handle slow changes in the noise statistics), is that each neuron must sample over extended periods of time to estimate its own mean and variance. If every neuron could access this statistic instantaneously, rapid compensation for the effects of noise on computation might be possible.

While such an instantaneous estimate of noise statistics would be impossible for a single neuron, neural networks consist of vast numbers of interacting cells. Can we estimate instantaneous noise level from the statistics of population activity? (This approach, of substituting a sample over time with a sample over neurons, is loosely inspired by the work of Mastrogiuseppe, Ostijic, and colleagues.)

One candidate is the average rate of activity across a large population of neurons. For binary (and spiking) neural networks, the mean population activity is equivalent to the sparsity in spiking activity. Homeostatic mechanisms that detect and maintain a target level of sparsity might stabilize neural computations against fluctuating noise statistics.

Inhibitory interneurons play a role in achieving stable levels of network activity, and also are responsible for oscillations. One might therefore conjecture that feedback about population activity level from inhibitory cells, and the oscillations introduced by this feedback, could play a role in making computation robust to noise.

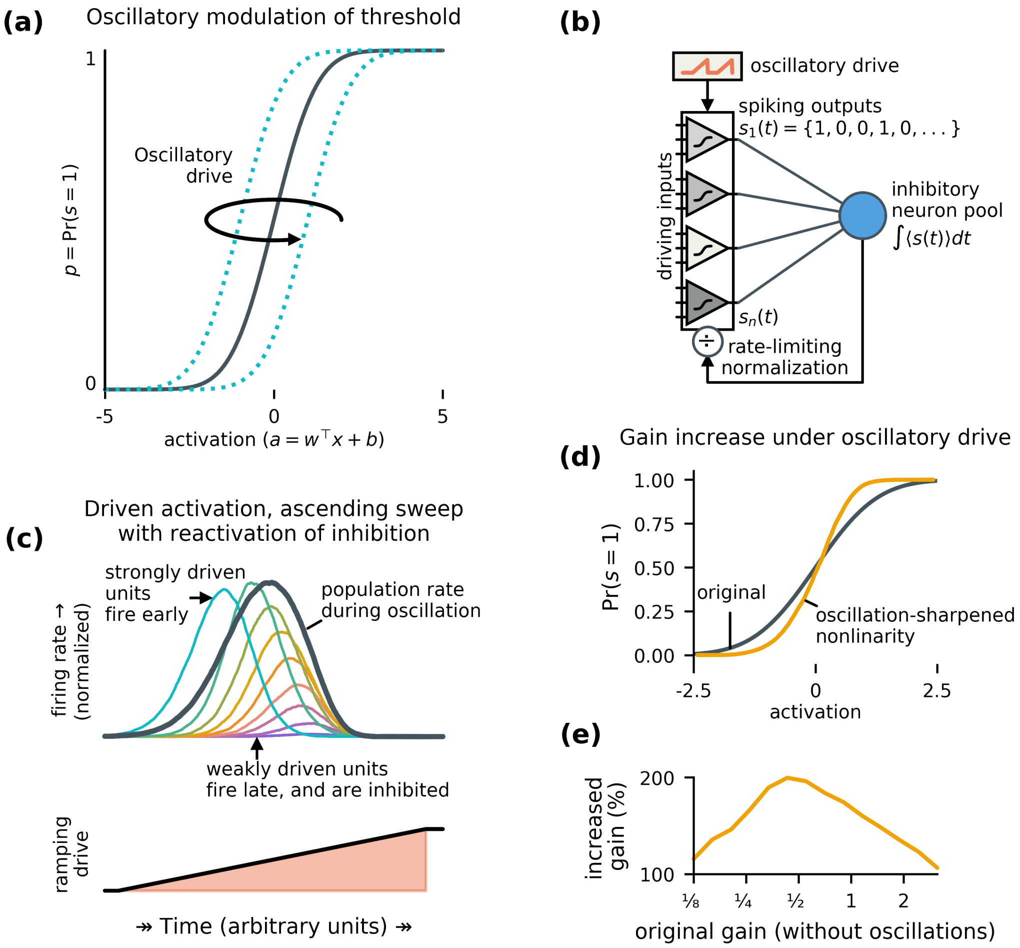

Figure 5: oscillatory drive can increase gain in a stochastic binary network (a) Oscillatory drive can be viewed as a form of shared threshold variation. During the ascending phase of an oscillation, neurons are released from inhibition. Those receiving more excitatory drive will fire earlier. **(b)** If oscillatory drive is combined with inhibitory feedback, strongly-driven (early-firing) units inhibit weakly-activated neurons. (shaded colors indicate different levels of drive per neuron.) Here, we integrate the recent population activity, which is the average of the spiking outputs "s", and apply feedback inhibition so that the total number of spikes within one oscillation cycle is conserved. **(c)** Firing rate during the ascending phase of an oscillation. Colored curves reflect neurons with different levels of excitatory drive. Strongly-driven units (blue) fire early, while weakly-driven ones (violet) file late. Early-firing units recruit inhibition, which reduce the firing of late-firing ones. **(d)** Simulation of gain modulation due to oscillations and inhibitory feedback. We drive neurons with different amounts of input activation (sampled per neuron $a \sim \mathcal{N}(0,1)$), as well as a ramping drive. Each cell is refractory, and fires at most once per oscillation. Feedback inhibition limits the population rate so that only 50% of cells fire per oscillation cycle. The effective firing-rate nonlinearity is measured via Monte-Carlo simulation (50K replicas). **(e)** In this model, the biggest gain increase was observed for neurons with an initial nonlinearity of $p = \Phi(a/2)$

To explore potential roles for oscillations in moderating neuronal noise, we first consider a toy model, of a population of dichotomized-Gaussian neurons in discrete time (Fig 5ab), during a single period of an oscillation, modeled as an increasing ramping drive:

$$
\begin{aligned}
d(t) &= d_0 + t \cdot d_r \\
a_i(t) &= w_i^\top x + b_i + d(t) \\
p_i(t) &= \Phi(\gamma_0 a_i(t)) \cdot g(t) \cdot \left[1 - \textstyle\sum_0^{t-1} s_i(t)\right] \\
g(t) &= \begin{cases} r_{\max} - r(t) & r(t) < r_{\max} \\ 0 & \text{elsewise} \end{cases} \\
s_i(t) &\sim \text{Bernoulli}(p(t)) \\
r(t) &= \textstyle\sum_0^{t-1} \langle s_i(t) \rangle_i \\
t &\in [0, T].
\end{aligned}
\tag{6}
$$

Here, $a(t)$, $p(t)$, and $s(t)$ are time-dependent vectors representing the neuronal activation, spiking probabilities, and spiking outputs, respectively. The activation a is a sum of the individual inputs, $w^\top x + b$, and a shared drive d that increases over time. Parameters d_0 and d_r control the initial value of the drive and the rate at which it increases. Firing probabilities are taken as a nonlinear function activation, in this case the CDF of the standard normal distribution Φ, with an additional (constant) gain parameter γ_0.

The firing probabilities are multiplied by a gating term $g(t)$, which limits the total average firing rate per oscillation cycle to be lest than $r_{\max} \in [0, 1]$. The exact form of this cutoff is not important, only that it terminate activity once a certain number of neurons in the population have fired.

This gating can be interpreted as a form of inhibition, which integrates the population rate so far in each oscillation cycle $r(t)$ and provides inhibitory feedback. The time reflects activity during the ascending phase of oscillation, and ranges from 0 to T. Each neuron is also refractory: spiking at most once per oscillation, reflected in the term $1 - \sum_0^{t-1} s_i(t)$.

In this network, strongly activated neurons require little additional drive to fire, and fire early in the rising phase of the oscillation. These spikes recruit inhibitory feedback that suppresses firing of weakly-activated neurons (Fig. 5bc). This interaction provides sharper contrast between strongly-driven and weakly-driven cells, which effectively amounts to a higher-gain in the transfer function and a reduction in neuronal variability. In simulation studies (Fig. 5e), we found that this effect could double the effective gain in some scenarios.

How is this possible? At first, it might seem surprising that we can remove noise from spiking activity in this way. After all, it is seldom possible to recover a lossless signal from one corrupted by noise. Noise in the stochastic binary neurons considered here arises from a single phenomenon: variability close to threshold.

Strongly driven or inhibited neurons are always far from threshold, and noise—whether it be from input, threshold variability, or intrinsic sources, is irrelevant. We are then mainly concerned with with cells that are slightly below (or above) threshold, which could be pushed to spike (or not) by extraneous inputs or fluctuations.

When we simulate a population of stochastic binary neurons responding to a ramping drive, we sample many repeated Bernoulli trials for each neuron. The activity over one sweep of the oscillation therefore (indirectly) reflects an average of many Bernoulli trials, and this allows us to remove some of the stochasticity related to threshold noise. The rising sweep of activating drive (which may reflect either excitation, or removal or recovery from inhibition) ensures that neurons fire in order from most to least activated.

The global inhibitory mechanism ensures that (on average) cells with the top top k% of activating drive are the ones that spike. This k-winners-take-all property has been explored extensively, as a generically useful computation in neural networks (e.g. Maass 2000).

0.0.4 4. Continuous-time spiking models

So far, we explored noise in stochastic binary networks in terms of a Gaussian model of threshold variability. In this model, we saw that noise could modulate firing rate nonlinearities, which could be computationally useful if it could be controlled. We also saw that driving oscillations could control effective noise levels.

We now consider a direct translation of discrete-time stochastic binary networks to continuous time Poisson spiking networks. The continuous time limit of a Bernoulli neuron is an inhomogeneous Poisson point-process model. Such models have been used extensively in analysis of spiking neural data sets, where they are terms auto-regressive Point-Process Generalized Linear Models (PP-GLMs). PP-GLMs are a useful starting point for building models of neural dynamics, and represent a compromise between mathematical simplicity and biological realism (see Ostojic and Brunel (2011), Truccolo (2016) for a review).

In the PP-GLM framework, we divide each of the time-steps in the discrete-time Bernoulli model into progressively smaller bins, preserving the average firing rate. In the limit of infinitesimal time bins, spiking is represented by a time-varying rate $\lambda(t) \approx p_t$ (Truccolo et al. 2005):

$$\begin{aligned}\Pr(k \text{ spikes} \in t \ldots t+\Delta t)& \\ &\sim \text{Poisson}\left(\int_t^{t+\Delta t} \lambda(t) dt\right) \\ &\approx \Delta t \cdot \lambda(t) \\ \lambda(t) &= \exp(a(t)).\end{aligned} \tag{7}$$

The Poisson and Bernoulli models behave similarly at low firing rates, but behave very differently at high rates. In the Bernoulli case, firing variability is suppressed for large rates, near $p{\approx}1$. For a Poisson process, however, the variance grows linearly in time and with firing rate.

We illustrate this in Figure 6, in which we take a Bernoulli network trained to model a target function (Fig. 4, Fig. 6a) and sample it in continuous time as a Poisson process (Fig. 6b). To match the continuous-time Poisson model to the discrete time Bernoulli model, we equate one second to one time step, and interpret the spiking probability p as a firing rate in Hz.

The larger variability of a Poisson process at high rates adds considerable noise to the output neuron (Fig. 6b). The more reliable behavior of the discrete-time Bernoulli model can be recovered, at least approximately, by including an absolute refractory period in the Poisson model (Fig. 6c). This refractory window here is equal to one time-bin (one second in the units chosen here).

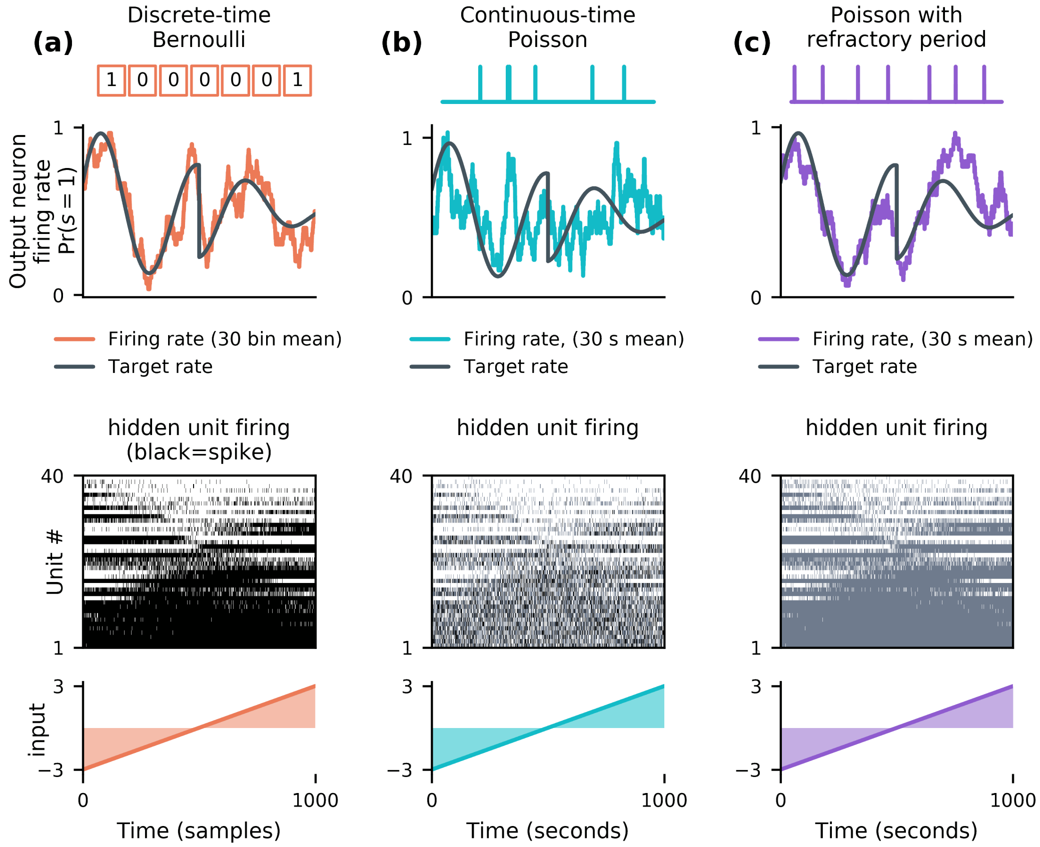

Figure 6: Refractoriness attenuates variability in continuous-time models (a) We trained a discrete-time Bernoulli model to generate a target function output, as in Figure 4. **(b)** The spiking output exhibits excess variability in the continuous-time Poisson case, since the Poisson model has large variance at high rates. **(c)** Adding an absolute refractory period limits the spiking variability within a single refractory time-window, recovering the original behavior. The nonlinearity in the refractory Poisson model differs from the ones used to train the original network, so we have adjusted the gain and bias of the nonlinearity to recover the original behaviot.

0.0.5 Conclude

In these notes, we have reviewed some implications for noise in simple models of neural computation. We reviewed a simple model, Gaussian noise interacting with a binary threshold. We explored some toy models for computational functions of noise (and correlations therein), including gain modulation and gating of neural interactions based on noise correlation structures. We then examined how oscillations would interact with threshold noise, and showed that oscillations with inhibition could attenuate noise.

Learning that occurs in the presence of noise can incorporate noise into the learned computation. This, however, assumes that properties of noise remain stable over time. In some scenarios, homeostasis might be able to provide this stability. On faster timescales, some computations, like K-winner-take-all (mediated by population oscillations), might also provide a “normalizing” effect. This confers some robustness to changing population statistics, and to some extent resembles the “batchnorm” procedure now widely used in training ANNs. Limiting spike-timing variability (and thereby noise) via refractoriness and inhibition is also important for attenuating variability and its impact on computations.

0.0.6 Cited

Ale, Angelique, Paul Kirk, and Michael PH Stumpf. 2013. "A General Moment Expansion Method for Stochastic Kinetic Models." The Journal of Chemical Physics 138 (17). AIP: 174101.

Bethge, Matthias, and Philipp Berens. 2008. "Near-Maximum Entropy Models for Binary Neural Representations of Natural Images." In Advances in Neural Information Processing Systems, 97–104.

Bradbury, James, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. 2018. JAX: Composable Transformations of Python+NumPy Programs (version 0.1.55). http://github.com/google/jax.

Byrne, Áine, Daniele Avitabile, and Stephen Coombes. 2019. "Next-Generation Neural Field Model: The Evolution of Synchrony Within Patterns and Waves." Physical Review E 99 (1). APS: 012313.

Cox, David R, and Nanny Wermuth. 2002. "On Some Models for Multivariate Binary Variables Parallel in Complexity with the Multivariate Gaussian Distribution." Biometrika 89 (2). Oxford University Press: 462–69.

Drezner, Zvi, and George O Wesolowsky. 1990. "On the Computation of the Bivariate Normal Integral." Journal of Statistical Computation and Simulation 35 (1-2). Taylor & Francis: 101–7.

Echeveste, Rodrigo, Laurence Aitchison, Guillaume Hennequin, and Máté Lengyel. 2019. "Cortical-Like Dynamics in Recurrent Circuits Optimized for Sampling-Based Probabilistic Inference." bioRxiv. Cold Spring Harbor Laboratory, 696088.

Emrich, Lawrence J, and Marion R Piedmonte. 1991. "A Method for Generating High-Dimensional Multivariate Binary Variates." The American Statistician 45 (4). Taylor & Francis: 302–4.

Ermentrout, G Bard, Roberto F Galán, and Nathaniel N Urban. 2008. "Reliability, Synchrony and Noise." Trends in Neurosciences 31 (8). Elsevier: 428–34.

Faisal, A Aldo, Luc PJ Selen, and Daniel M Wolpert. 2008. "Noise in the Nervous System." Nature Reviews Neuroscience 9 (4). Nature Publishing Group: 292.

Frostig, Roy, Matthew James Johnson, and Chris Leary. 2018. "Compiling Machine Learning Programs via High-Level Tracing." SysML.

Galán, Roberto F, Nicolas Fourcaud-Trocmé, G Bard Ermentrout, and Nathaniel N Urban. 2006. "Correlation-Induced Synchronization of Oscillations in Olfactory Bulb Neurons." Journal of Neuroscience 26 (14). Soc Neuroscience: 3646–55.

Hennequin, Guillaume, and Máté Lengyel. 2016. "Characterizing Variability in Nonlinear Recurrent Neuronal Networks." arXiv Preprint arXiv:1610.03110.

Keeley, Stephen L, David M Zoltowski, Yiyi Yu, Jacob L Yates, Spencer L Smith, and Jonathan W Pillow. 2019. "Efficient Non-Conjugate Gaussian Process Factor Models for Spike Count Data Using Polynomial Approximations." arXiv Preprint arXiv:1906.03318.

Maass, Wolfgang. 2000. "On the Computational Power of Winner-Take-All." Neural Computation 12 (11). MIT Press: 2519–35.

Macke, Jakob H, Manfred Opper, and Matthias Bethge. 2009. "The Effect of Pairwise Neural Correlations on Global Population Statistics." Max Planck Institute for Biological Cybernetics.

———. 2011. “Common Input Explains Higher-Order Correlations and Entropy in a Simple Model of Neural Population Activity.” Physical Review Letters 106 (20). APS: 208102.

Marder, Eve, and Astrid A Prinz. 2002. “Modeling Stability in Neuron and Network Function: The Role of Activity in Homeostasis.” Bioessays 24 (12). Wiley Online Library: 1145–54.

McDonnell, Mark D, and Derek Abbott. 2009. “What Is Stochastic Resonance? Definitions, Misconceptions, Debates, and Its Relevance to Biology.” PLoS Computational Biology 5 (5). Public Library of Science: e1000348.

McDonnell, Mark D, Nigel G Stocks, Charles EM Pearce, and Derek Abbott. 2008. “Stochastic Resonance.” Stochastic Resonance, by Mark D. McDonnell, Nigel G. Stocks, Charles EM Pearce, Derek Abbott, Cambridge, UK: Cambridge University Press, 2008.

Miller, Paul, and Jonathan Cannon. 2019. “Combined Mechanisms of Neural Firing Rate Homeostasis.” Biological Cybernetics 113 (1-2). Springer: 47–59.

Moss, Frank, Lawrence M Ward, and Walter G Sannita. 2004. “Stochastic Resonance and Sensory Information Processing: A Tutorial and Review of Application.” Clinical Neurophysiology 115 (2). Elsevier: 267–81.

O’Leary, Timothy. 2018. “Homeostasis, Failure of Homeostasis and Degenerate Ion Channel Regulation.” Current Opinion in Physiology 2. Elsevier: 129–38.

O’Leary, Timothy, and Eve Marder. 2016. “Temperature-Robust Neural Function from Activity-Dependent Ion Channel Regulation.” Current Biology 26 (21). Elsevier: 2935–41.

O’Leary, Timothy, Alex H Williams, Jonathan S Caplan, and Eve Marder. 2013. “Correlations in Ion Channel Expression Emerge from Homeostatic Tuning Rules.” Proceedings of the National Academy of Sciences 110 (28). National Acad Sciences: E2645–E2654.

Ostojic, Srdjan, and Nicolas Brunel. 2011. “From Spiking Neuron Models to Linear-Nonlinear Models.” PLoS Computational Biology 7 (1). Public Library of Science: e1001056.

Pearson, Karl. 1909. “On a New Method of Determining Correlation Between a Measured Character a, and a Character B, of Which Only the Percentage of Cases Wherein B Exceeds (or Falls Short of) a Given Intensity Is Recorded for Each Grade of a.” Biometrika 7 (1/2). JSTOR: 96–105.

Pillow, Jonathan W, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli. 2008. “Spatio-Temporal Correlations and Visual Signalling in a Complete Neuronal Population.” Nature 454 (7207). Nature Publishing Group: 995.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. 2014. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models.” arXiv Preprint arXiv:1401.4082.

Rule, Michael E., David Schnoerr, Matthias H. Hennig, and Guido Sanguinetti. 2019. “Neural Field Models for Latent State Inference: Application to Large-Scale Neuronal Recordings.” PLOS Computational Biology 15 (11). Public Library of Science: 1–23. https://doi.org/10.1371/journal.pcbi.1007442.

Rule, Michael, and Guido Sanguinetti. 2018. “Autoregressive Point Processes as Latent State-Space Models: A Moment-Closure Approach to Fluctuations and Autocorrelations.” Neural Computation 30 (10). MIT Press: 2757–80.

Schnoerr, David, Guido Sanguinetti, and Ramon Grima. 2014. “Validity Conditions for Moment Closure Approximations in Stochastic Chemical Kinetics.” The Journal of Chemical Physics 141 (8). AIP: 08B616_1.

———. 2015. “Comparison of Different Moment-Closure Approximations for Stochastic Chemical Kinetics.” The Journal of Chemical Physics 143 (18). AIP Publishing: 11B610_1.

———. 2017. “Approximation and Inference Methods for Stochastic Biochemical Kinetics—a Tutorial Review.” Journal of Physics A: Mathematical and Theoretical 50 (9). IOP Publishing: 093001.

Truccolo, Wilson. 2010. “Stochastic Models for Multivariate Neural Point Processes: Collective Dynamics and Neural Decoding.” In Analysis of Parallel Spike Trains, 321–41. Springer.

———. 2016. “From Point Process Observations to Collective Neural Dynamics: Nonlinear Hawkes Process Glms, Low-Dimensional Dynamics and Coarse Graining.” Journal of Physiology-Paris 110 (4). Elsevier: 336–47.

Truccolo, Wilson, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. 2005. “A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects.” Journal of Neurophysiology 93 (2). American Physiological Society: 1074–89.

Wu, Yue, Keith B Hengen, Gina G Turrigiano, and Julijana Gjorgjieva. 2019. “Homeostatic Mechanisms Regulate Distinct Aspects of Cortical Circuit Dynamics.” bioRxiv. Cold Spring Harbor Laboratory, 790410.

Zenke, Friedemann, and Wulfram Gerstner. 2017. “Hebbian Plasticity Requires Compensatory Processes on Multiple Timescales.” Philosophical Transactions of the Royal Society B: Biological Sciences 372 (1715). The Royal Society: 20160259.

Zenke, Friedemann, Guillaume Hennequin, and Wulfram Gerstner. 2013. “Synaptic Plasticity in Neural Networks Needs Homeostasis with a Fast Rate Detector.” PLoS Computational Biology 9 (11). Public Library of Science: e1003330.

Zhou, Pengcheng, Shawn Burton, Nathan Urban, and G Bard Ermentrout. 2013. “Impact of Neuronal Heterogeneity on Correlated Colored Noise-Induced Synchronization.” Frontiers in Computational Neuroscience 7. Frontiers: 113.|