Novelty Detection, Insect Olfaction, Mismatch Negativity, and the Representation of Probability in the Brain

Terrence C. Stewart (terrence.stewart@nrc-cnrc.gc.ca)
National Research Council Canada, University of Waterloo Collaboration Centre
Waterloo, ON, Canada

P. Michael Furlong (michael.furlong@uwaterloo.ca)
Kathryn Simone (kathryn.simone@uwaterloo.ca)
Madeleine Bartlett (madeleine.bartlett@uwaterloo.ca)
Jeff Orchard (jorchard@uwaterloo.ca)
Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

Abstract

We present a unified model of how groups of neurons can represent and learn probability distributions using a biologically plausible online learning rule. We first present this in the context of insect olfaction, where we map our model onto a well-known biological circuit with a single output neuron that represents whether the current stimulus is novel or not. We show that the model approximates a Bayesian inference process, providing an explanation as to why the current flowing into the output neuron is proportional to the expected probability of that stimulus. Finally, we extend this model to show that the same circuit can detect deviations in temporal patterns, like the expectation violations that elicit the EEG mismatch negativity signal.

Keywords: novelty detection; insect olfaction; mismatch negativity; neural representation; hyperdimensional computing; fractional binding; spatial semantic pointers; Bayesian inference

Introduction

It is critically important for any cognitive agent to recognize their sensory stimuli as novel or unexpected. Different strategies may be applicable, depending on whether one is in a familiar situation (in which case one can safely rely on previously learned knowledge), or in a novel situation (in which case a more careful and exploratory strategy may be appropriate). In mammals, an example of this is seen in the Mismatch Negativity signal, a strong EEG signal that appears approximately 200ms after a surprising stimulus (Pazo-Alvarez et al., 2003). This seems to be an automatic process, occurring regardless of whether the participant is paying attention to the stimulus or not. This automaticity and the speed of the response suggests that this novelty detection is a simple and basic process that may be understood without involving the entire brain.

Furthermore, a specific neural circuit for novelty detection has also been identified in the insect brain. The MBON (Mushroom Body Output Neuron) α'3 neuron is consistently active in the presence of novel odours, and is silent for odours that have been previously encountered. The inputs to this neuron come from Kenyon Cells, which form a very sparse representation of the current odour, so each odour corresponds to a different (sparse) pattern of activity in these neurons. In (Dasgupta et al., 2018), this system is compared to the computer-science idea of a Bloom Filter, a type of hashtable where input data is converted into a sparse representation, and then individual elements of that representation (i.e. the activity of the Kenyon Cells) are used to quickly determine whether the current input is likely to be novel or not. The core idea is to do this without requiring a complete database of every odour that has been previously observed; instead, use the overlap in the sparse representation as a fast estimate as for the input's novelty.

In this paper, we present a simple model of this novelty detection system that is compatible with the above idea, but interprets the computation being performed by the neurons in a slightly different way. In particular, we suggest that the neurons (and the connection weights between them) are in fact representing a probability distribution, and "novelty" is detected if the current input is highly unlikely according to that distribution. We show that a very simple learning rule, combined with a particular method for encoding information in neurons, results in a network that accurately estimates the observed probability distribution of different inputs, and that a single neuron (such as the MBON $\alpha'3$ neuron) can use this distribution to signal novelty.

Given this insect-based model, we then expand the system to encode information over time, and show that the very same model is capable of detecting the sort of temporal novelty that is the hallmark of the Mismatch Negativity signal in mammals. This expanded system makes use of Legendre Memory Units (LMUs), a recurrent neural structure that has been mapped to Time Cells (Voelker et al., 2019), temporal patterns in the cerebellum (Stöckel et al., 2021), and has been shown to improve performance on Machine Learning tasks over LSTMs (Voelker et al., 2019) and Transformers (Chilkuri et al., 2021).

Vector Representation

Given the wide range of possible inputs for a novelty detection system, we define our inputs simply as vectors. These can be of arbitrary dimensionality, and we do not constrain their magnitude. In this way we can deal with inputs such as odours (which in insects can be thought of as vectors in a 50-dimensional space based on 50 different odour-sensing neurons), or images, or numerical values in general. Interestingly, the resulting model we define here will not need to be changed in any way to handle different types of inputs, other

than using our general-purpose input mapping.

When inputting a numerical vector into a neural network, it is generally useful to transform it in some way. Typically, this is some sort of normalization, ensuring that the input has a mean of zero and a standard deviation of one, for example. However, normalizing requires knowing the overall range of possible input values. As an alternative, we adapt a method used for encoding arbitrary-length lists into a fixed-length vector.

In particular, Plate (1995) suggests using randomly chosen high-dimensional unit vectors for representing structured information. The list $[A,B,C]$ can be represented as the vector $A \circledast X + B \circledast X \circledast X + C \circledast X \circledast X \circledast X$, where $\circledast$ is circular convolution, A, B, C, and X are randomly chosen unit vectors (and X is also *unitary*, ensuring that its magnitude stays 1 after repeated circular convolution). Since circular convolution is element-wise multiplication in the Fourier domain, the repeated convolution $X \circledast X \circledast X$ can be written as X^3, leading naturally to a generalization where the exponent is a real number instead of an integer.

In other words, we can represent the position x on the x-axis as X^x, where X is a randomly chosen unitary vector, and where the exponent means "take the Fourier transform, then raise each element to the power x, then take the inverse Fourier transform". For a two-dimensional input $< x,y >$ we can compute $X^x \circledast Y^y$, and so on for higher dimensions. Importantly, the resulting vector is *always* a D-dimensional unit vector (where D is the dimensionality of the base vectors X, Y, etc., which we set to be 1024 here). This is true regardless of the number of actual inputs, meaning that we do not need to change anything about the internals of our novelty detection model to deal with different inputs. Furthermore, since the input is always a unit vector, we do not need to further normalize the inputs.

While this approach to representation was mentioned in (Plate, 1995), it is further analyzed in (Lu et al., 2019). In particular, the Fourier transform of a unitary vector leads to a vector with complex components $[e^{i\phi_1}, e^{i\phi_2}, e^{i\phi_3}, ...]^{\mathrm{T}}$, so raising this to a power x gives $[e^{i\phi_1 x}, e^{i\phi_2 x}, e^{i\phi_3 x}, ...]^{\mathrm{T}}$. This produces a series of oscillations, and as long as the ϕ_i values are relatively prime, the exact pattern of oscillation will never repeat, no matter how much one varies x. Of course, for some x values the resulting vectors will be very close to each other, leading to the possibility of confusion between some points, but in a high-dimensional space (D), this will be uncommon.

This idea of vectors being close to each other also provides us with an important parameter for the model. In particular, a small change in x will produce a small change in the vector X^x. If we quantify this with the dot product, it can be shown that, in the limit as $D \to \infty$, the similarity between X^a and X^{a+x} approaches the normalized sinc function, $\sin(\pi x)/(\pi x)$. That is, for $x = 1$, the two vectors will be orthogonal (no similarity), but for smaller values of x the vectors will be closer and closer to each other. This gives the representation a particular *scale*. Depending on our inputs, we may want to control this scale, and we do this by introducing a length scale parameter λ, and encode information as $X^{x/\lambda} \circledast Y^{y/\lambda}$. In this way, values that are less than λ apart will yield vectors with high similarity. For our novelty detection system, this gives us a reference for how different an input needs to be from other inputs to be considered novel.

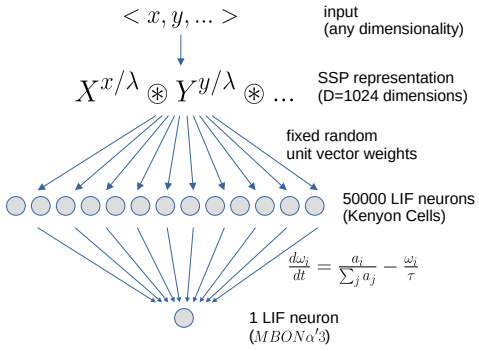

Figure 1: Our novelty detection model. Arbitrary input is converted into an SSP and passed to LIF neurons via randomly chosen fixed connection weights. Output weights are increased whenever a neuron is active, and decayed over time. In the insect, the neurons correspond to Kenyon Cells and the output is the MBON $\alpha'3$ neuron.

We call this style of representation a *Spatial Semantic Pointer*, or SSP, since it gives a compressed representation of an infinitely large space, but maintains semantic information in that it yields high similarity for nearby x values. This approach to continuous representation is first found in (Plate, 1992), and more recently in (Frady et al., 2018), where it is known as *Fractional Power Encoding*, or *Fractional Binding*, since the $\circledast$ operator is thought of as a binding operator in Vector Symbolic Architectures.

Computational Model

Our computational model is shown in Figure 1. The first step is to convert the input into an SSP vector, using the above approach. We use $D = 1024$ here. Next, we define 50,000 neurons, each with a separate randomly-chosen 1024-dimensional vector for its input connection weights. That is, each neuron receives as input the dot product between the actual input and a randomly chosen "preferred stimulus" for that neuron. This is a generalization of the standard finding of preferred-direction-vectors in sensory and motor cortices (Georgopoulos et al., 1982; Schwartz et al., 1988). Each neuron also has a negative bias input that controls how similar the input needs to be to its ideal stimulus in order for the neuron to be active. This controls the sparsity of the neural representation. While any rectified neuron model can be used for this, here we use spiking Leaky Integrate-and-Fire (LIF) neurons with a bias of 0.9.

Given this input, the 50,000 neurons will form a sparse representation of that input, corresponding to Kenyon Cells observed the insect. In order to learn what stimuli are common

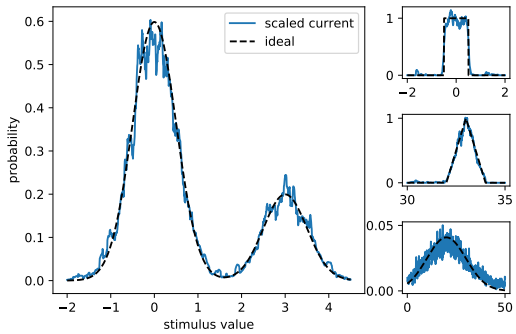

Figure 2: Queried probability after exposure to four different distributions of data. Blue line is the total current flowing into the output neuron, linearly scaled to unit area. Black dashed line is the ideal distribution. Four different distributions are shown. Note that even though the x-axis is very different for each distribution, the model itself is not changed in any way.

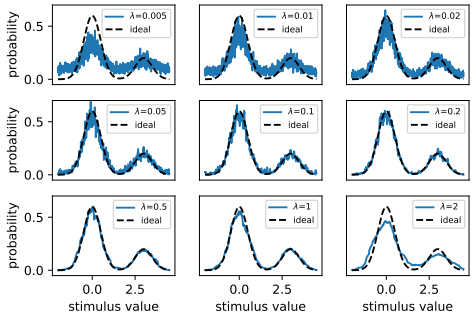

Figure 3: Queried probability given exposure to a particular distribution of data. Nine different versions of the model are shown, each with a different length scale (λ). When λ is too small (< 0.05) or too large (> 1), the representation is less accurate, but is fairly robust in-between.

given this representation, we add a simple learning rule to the output connection weights of the model. These weights are initialized to zero, and each is increased proportional to the normalized activity of its corresponding, pre-synaptic neuron. Finally, we also decay the weights toward 0 over time, resulting in a weight learning rule of $\frac{d\omega_i}{dt} = \frac{a_i}{\sum_j a_j} - \frac{\omega_i}{\tau}$, where a_i is the activity of the i^{th} neuron, ω_i is the connection weight, and τ is a time constant for the decay.

The overall output from this system should be large for familiar inputs and small for unfamiliar inputs. Surprisingly, as shown in Figure 2, the output current *is proportional to the probability of the input!* That is, rather than just detecting novel vs familiar inputs, the system learns to directly represent the probability distribution of the input. Here we present inputs sampled from four different distributions (black dashed lines), and then measure the output current over a range of values from across the input domain. Importantly, this method automatically calibrates itself for whatever range of input values it receives. For example, the triangular distribution consists of values between 32 and 34, while the square and bimodal distributions cover values in the smaller ranges of -2 to 5). This is because the neurons have preferred inputs that cover the entire 1024-dimensional space, which itself covers the entire infinite range of possible inputs to the model, thanks to the SSP representation.

However, the wide distribution (Figure 2, bottom-right) between 0 and 50 shows higher variance in the probability representation. This is due to the length-scale parameter λ. The effects of this parameter are explored in Figure 3, showing that high variance can result from a value that is too small, but a value that is too large can lead to reduced accuracy. Thus, while this approach is robust to values anywhere on the x-axis, it is sensitive to the overall scale of values being represented.

We can also represent multidimensional probability representations just by encoding samples as $X^x \circledast Y^y$. Again, nothing is changed about the neural aspect of the model; all that is

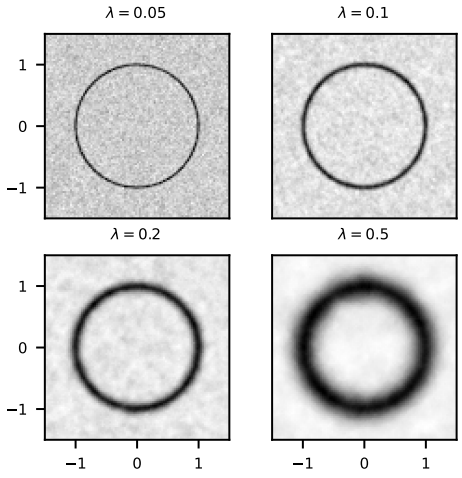

Figure 4: Queried probability given exposure to a two-dimensional data sampled evenly from a unit circle. Four different versions of the model are shown, each with a different length scale (λ). With larger λ the model generalizes across a larger region, so points slightly off the unit circle are not considered novel.

changed is the mapping into the 1024-dimensional SSP space. Figure 4 shows the resulting probability distribution (the blue curve in Figures 2 and 3) for two-dimensional input data that is sampled evenly from the unit circle. Notice that the length scale λ controls the resolution of the representation, controlling how far off the unit circle an input needs to be before it is considered to be different enough from observed data to be novel.

For a more detailed analysis of the accuracy of this model, Figure 5 demonstrates the overall linearity of the representation, and Figure 6 shows the accuracy as the number of neurons and length scale λ are varied.

Why This Works

To understand why this works, we have to consider the embedding of the SSP representation. The dot product be-

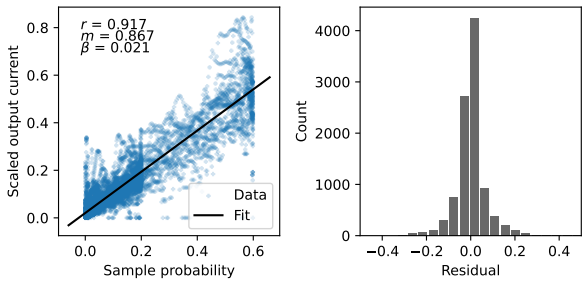

Figure 5: Characterization of relationship between sample probability and output current. Left: Scatterplot depicting individual samples of output current against observation probabilities given the distribution, with data merged across 10 runs of the model. Apparent is a linear relationship (regression line shown in black) and uniform variance in the representation error as a function of probability. Results shown are for $N = 1000$ neurons, length scale $\lambda = 0.2$. Right: Histogram of residuals between true and estimated distribution pooled across 10 runs of the model. The mean of errors is not different from zero ($P > 0.999$, two-sided one-sample t-test).

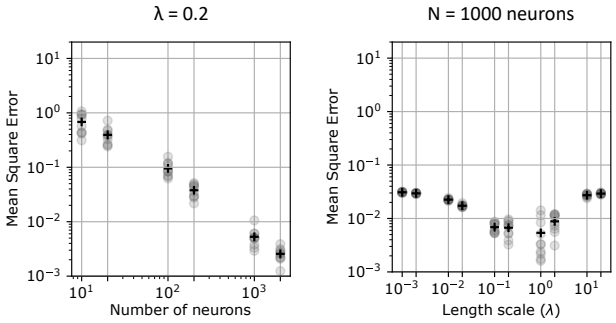

Figure 6: Representation error as a function of model parameters. Left: Representation error falls with the number of neurons in the model. Each gray dot represents one independent run of the model, and the black markers indicate the mean error across runs for that condition. Right: λ has a modest effect on representation error. Representation error is higher for both small and large values of λ. Intermediate values of λ result in more variable performance but generally lower error, indicated by a larger vertical spread of performance.

tween SSP representations, X^x and $X^{x'}$, approximates a quasi-kernel function, in this case the normalized sinc function, $X^x \cdot X^{x'} \approx \text{sinc}(\|x-x'\|) = \sin(\pi\|x-x'\|)/(\pi\|x-x'\|)$. Since the sinc function is an admissible kernel for kernel density estimation (Tsybakov, 2009, §1.3), we can view the dot product between an SSP encoded-vector and other vectors as approximating a probability. The argument is as follows:

Consider a randomly chosen unit vector, $\mathbf{w}_i$, of synaptic weights feeding into the i^{th} neuron of a network. The weights $\mathbf{w}_i$ will have some similarity with X^x, although for randomly chosen $\mathbf{w}_i$, it is likely to be small. Consequently, we can consider any synaptic weight matrix as being the sum of a vector that is orthogonal to X^x, and one that is a sum of a (possibly empty) set of points encoded using the SSPs. That is, $\mathbf{w}_i = \frac{1}{n_i}\sum_{k=1}^{n_i} X^{x_k} + w_{\text{orthogonal}}$.

We can then consider the input current of any given neuron as being an approximation of the probability of the input point, conditioned on a binary variable:

$$\begin{aligned}\mathbf{w}_i \cdot X^x &= \frac{1}{n_i}\sum_{k=1}^{n_i} X^x \cdot X^{x_k} + w_{\text{orthogonal}} \cdot X^x \\ &\approx \frac{1}{n_i}\sum_{k=1}^{n_i} \text{sinc}(\|x-x_k\|) \\ &\approx P(X = x \mid V_i)\end{aligned}$$

where the distribution conditioned on variable V_i is defined by the sinc kernel and the points $\{x_1, \ldots, x_{n_i}\}$. If we assume a rectified linear neuron, $a_i = \text{ReLU}(\mathbf{w}_i \cdot X^x - b_i)$, with a bias, b_i chosen according to the method identified by Glad et al. (2003), then the activity of a neuron is exactly a probability. When we normalize the population's activities, $\hat{a}_i = a_i / \sum_j a_j$, we see that this population is conducting Bayesian inference on the variables, assuming a uniform prior over $P(V_i)$, as shown below.

$$\begin{aligned}P(V_i \mid X = x) &= \frac{P(X = x \mid V_i)P(V_i)}{P(X = x)} \\ &= \frac{P(X = x \mid V_i)P(V_i)}{\sum_j P(X = x \mid V_j)P(V_j)}\end{aligned}$$

If we let $P(V_i)$ be a non-informative prior, then we can remove it from the equation, yielding

$$P(V_i \mid X = x) = \frac{P(X = x \mid V_i)}{\sum_j P(X = x \mid V_j)} \approx \frac{a_i}{\sum_j a_j} = \hat{a}_i$$

which is the normalized neuron activity.

While the above analysis assumes a ReLU neuron, we note that LIF neurons, when averaged over time, produce an output that is fairly similar to a ReLU, other than the saturation behaviour. This saturation reduces the neuron output at high similarity values, but since it is still monotonically increasing (and non-negative), the LIF neuron's overall firing rate will still preserve the properties of the kernel function that make them suitable for density estimation.

Once normalized, we know that their firing rates will always be scaled between 0 and 1. Consequently, the firing rates of the neurons in the population can be used as the bases in a reproducing kernel Hilbert space (for a good short introduction see Ghojogh et al., 2021). Thus, they can approximate the probability density function over the points, encoded in the activity of those neurons whose input weights are not orthogonal to the manifold $X^x, \forall x \in \mathbb{R}$. Our learning rule then simply learns the appropriate weights, α_i for the expression

$$P(X = x) = \sum_i^n \alpha_i k'(x, x_i) \tag{1}$$

where $k'(x, x_i) \approx LIF(\text{sinc}(\|x - x_i\|) - b_i)$, and x_i is a solution to $\arg\min_x \|1 - \mathbf{w}_i \cdot X^x\|_2^2$. Appealing to the representer theorem (Schölkopf et al., 2001), then there exists an optimal approximation of the probability distribution being learned, constrained by the kernel function induced by the SSP encoding, neural activation function, and the implicit collection

of points in the domain sampled by the synaptic weights. Depending on how the synaptic weights are chosen, better or worse approximations can be found.

Mismatch Negativity

The theoretical argument and simulations above establish that this system is able to represent multi-dimensional probability distributions, and then detect when a stimulus is a low-probability event. However, this is all based on the instantaneous input to the model. What if we also want to detect novel *temporal* patterns? This would be needed to, for example, respond to a stimulus being presented for an unexpected length of time, or at a different frequency. The well-known Mismatch Negativity signal is observed for exactly these sorts of novel stimuli. However, the model as presented so far, is only producing output based on the current input, and so cannot be sensitive to such temporal differences.

To address this problem, we need a way to take an input value that changes over time and convert it into a multi-dimensional value that represents the recent history of that signal. Fortunately, a method for doing this already exists: Voelker et al. (2019) presented a linear differential equation ($\frac{dm}{dt} = Am + Bx$) that converts an input signal x into a vector m which encodes the recent history of x as a set of coefficients of Legendre polynomials. This LMU (Legendre Memory Unit) and the associated LDN (Legendre Delay Network) can be implemented in spiking neurons, resulting in neural activity corresponding to Time Cells (Voelker et al., 2019), and has also been shown to out-perform LSTMs, GRUs, and Transformers on standard machine learning benchmarks that require temporal information (Chilkuri et al., 2021; Voelker et al., 2019).

With this in mind, we can construct a version of our model that responds to temporal signals by passing input data into an LDN to create the Legendre representation of the input, and then feeding that representation into the same novelty detection system defined above. As before, we do not need to change anything about the model to handle the increased dimensionality of the input.

The results for an input pattern that starts as a 1Hz signal, then switches to a 2Hz signal, and then back again are shown in Figure 7. We use a two-dimensional Legendre representation to encode the previous 2 seconds of the input (second graph). We also set the weight decay on the connection weights to $\tau = 5$ seconds. If we now feed the temporal pattern into the novelty detection system, the probability estimate (third graph) shows an increasing estimate of probability as the 1Hz signal becomes more familiar, and then a sudden drop in the probability estimate when it switches to a 2Hz signal, and then another drop when it returns to 1Hz. If we connect this probability estimate output as an inhibitory signal to a single spiking Leaky Integrate-and-Fire neuron (corresponding to the MBON $\alpha'3$ neuron), then we can see this neuron firing when the temporal pattern changes (fourth graph). Figure 8 shows the same result, but for a different input pattern

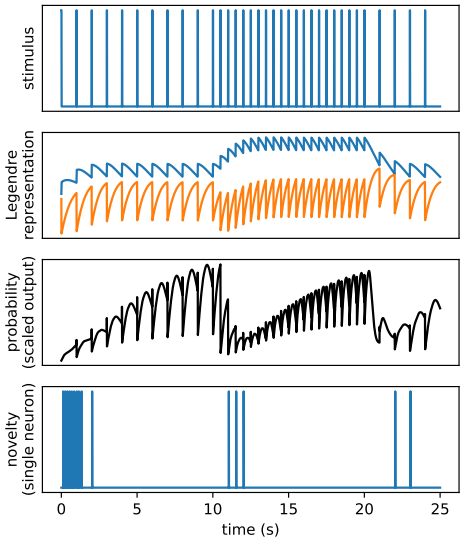

Figure 7: Detection of temporal novelty with the same circuit. The stimulus (first graph) is a regular pattern which changes frequency. The previous 2 seconds of the temporal input pattern is encoded using Legendre coefficients (second graph), and then fed into the same novelty detection model as before. When the output probability is low (third graph), novelty is detected (fourth graph).

with occasional outlier values.

Conclusions

We have demonstrated a simple single-hidden-layer neural network that learns to represent the probability distribution of its recent inputs. The output from this network can be used to detect novel inputs (the output is proportional to the learned probability, so it will be small if the learned probability for that input is low). Furthermore, the model works for different input dimensionalities and ranges, as inputs are converted into points on a D-dimensional sphere no matter what dimensionality those inputs are originally. The main parameter affecting performance is the chosen length scale λ, which does need to be tuned (Figure 6). Increasing the number of neurons improves performance, and the only other parameters are D (the dimensionality of the SSP space) and the neuron bias parameter, which controls the sparsity of the representation. Characterizing the effects of these parameters is ongoing.

Furthermore, we have extended this model to detecting temporal novelty as well, by exploiting a separate neural system (the LMU) to convert an input signal into a vector that represents the recent history of that signal. This gives a potential mechanism for detecting novelty that could trigger the observed Mismatch Negativity signal. This is somewhat surprising, in that our original model was inspired by the insect mushroom body system, while Mismatch Negativity is observed in mammals. Our ongoing work is to further investigate parallels between these two systems. However, it should

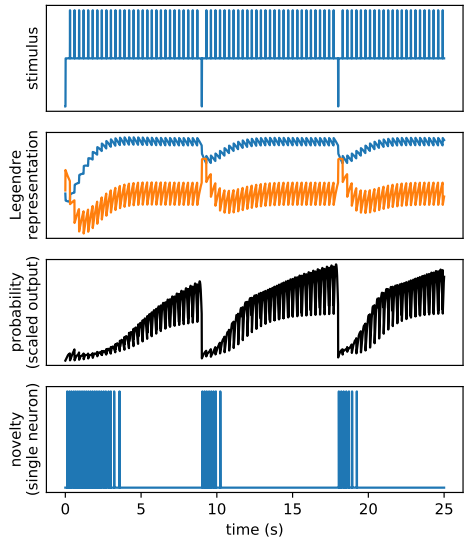

Figure 8: Detection of temporal novelty with the same circuit. The stimulus (first graph) is a regular pattern with occasional rare inputs (e.g. auditory tones of one frequency with occasional tones of a different frequency). The previous 2 seconds of the temporal input pattern is encoded using Legendre coefficients (second graph), and then fed into the same novelty dection model as before. When the output probability is low (third graph), novelty is detected (fourth graph).

be noted that the current model cannot directly explain the Mismatch Negativity signal, since nothing in the current system would generate a large and coherent EEG signal. That said, we do believe our model could act as a trigger telling the brain that a novel stimulus has occurred, which then leads to some other brain mechanism coming online which does generate the large change in electric field that is detected as Mismatch Negativity.

Acknowledgments

This project was supported by collaborative research funding from the National Research Council of Canada's Artificial Intelligence for Logistics program.

References

Chilkuri, N., Hunsberger, E., Voelker, A., Malik, G., & Eliasmith, C. (2021). Language modeling using lmus: 10x better data efficiency or improved scaling compared to transformers. *arXiv preprint*.

Dasgupta, S., Sheehan, T., Stevens, C., & Navlakha, S. (2018). A neural data structure for novelty detection. *Proceedings of the National Academy of Sciences*, *115*, 201814448.

Frady, E. P., Kanerva, P., & Sommer, F. T. (2018). A framework for linking computations and rhythm-based timing patterns in neural firing, such as phase precession in hippocampal place cells. *Conference on Cognitive Computational Neuroscience*, 1–5.

Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., & Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, *2*, 1527–1537.

Ghojogh, B., Ghodsi, A., Karray, F., & Crowley, M. (2021). Reproducing kernel Hilbert space, Mercer's theorem, eigenfunctions, Nyström method, and use of kernels in machine learning: Tutorial and survey. *arXiv preprint arXiv:2106.08443*.

Glad, I. K., Hjort, N. L., & Ushakov, N. G. (2003). Correction of density estimators that are not densities. *Scandinavian Journal of Statistics*, *30*(2), 415–427.

Lu, T., Voelker, A. R., Komer, B., & Eliasmith, C. (2019). Representing spatial relations with fractional binding. *Proc. of the Cognitive Science Society*.

Pazo-Alvarez, P., Cadaveira, F., & Amenedo, E. (2003). MMN in the visual modality: A review. *Biological Psychology*, *63*(3), 199–236.

Plate, T. A. (1992). Holographic recurrent networks. In S. Hanson, J. Cowan, & C. Giles (Eds.), *Advances in neural information processing systems*. Morgan-Kaufmann.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*, 623–641.

Schölkopf, B., Herbrich, R., & Smola, A. J. (2001). A generalized representer theorem. *Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001 Amsterdam, The Netherlands, July 16–19, 2001 Proceedings 14*, 416–426.

Schwartz, A. B., Kettner, R. E., & Georgopoulos, A. P. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. i. relations between single cell discharge and direction of movement. *8*, 2913–2927.

Stöckel, A., Stewart, T. C., & Eliasmith, C. (2021). Connecting biological detail with neural computation: Application to the cerebellar granule-golgi microcircuit. *Topics in Cognitive Science*, *13*(3), 515–533.

Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer.

Voelker, A. R., Kajić, I., & Eliasmith, C. (2019). Legendre memory units: Continuous-time representation in recurrent neural networks. *Advances in Neural Information Processing Systems*, 15544–15553.