A Solution to the Dynamics of the Prescribed Error Sensitivity Learning Rule

Aaron Russell Voelker

Centre for Theoretical Neuroscience technical report. Oct 1, 2015

Abstract

Prescribed Error Sensitivity (PES) is a biologically plausible supervised learning rule that is frequently used with the Neural Engineering Framework (NEF). PES modifies the connection weights between populations of neurons to minimize an external error signal. We solve the discrete dynamical system for the case of constant inputs and no noise, to show that the decoding vectors given by the NEF have a simple closed-form expression in terms of the number of simulation timesteps. Moreover, with $\gamma = (1 - \kappa\|\mathbf{a}\|^2) < 1$, where κ is the learning rate and $\mathbf{a}$ is the vector of firing rates, the error at timestep k is the initial error times γ^k. Thus, $\gamma > -1$ implies exponential convergence to a unique stable solution, $\gamma < 0$ results in oscillatory weight changes, and $\gamma \leq -1$ implies instability.

1 Introduction

The Neural Engineering Framework (NEF), proposed by Eliasmith and Anderson (2003), is a method of constructing biologically plausible spiking networks. To build and simulate such models, the Centre for Theoretical Neuroscience makes extensive use of the open-source software, Nengo (Bekolay et al., 2014).

The NEF typically learns its connection weights offline, but Nengo also supports a number of biologically plausible supervised and unsupervised learning rules to learn these weights online. By far, the most commonly used learning rule in Nengo is the Prescribed Error Sensitivity (PES) rule, which learns a function by minimizing an external error signal (Bekolay et al., 2013). This learning rule has been used to model episodic memory (Trujillo and Eliasmith, 2014), hierarchical reinforcement learning (Rasmussen and Eliasmith, 2014), adaptive motor control (Komer, 2015), and many other tasks.

While the usage and overall "effect" of the learning rule is intuitively clear, several questions have not been explicitly addressed to date:

- Under what conditions is the rule guaranteed to converge to a unique optimal solution?
- How does the learning rate affect the rate of convergence?
- Can we be precise about the form of the weights as a function of time, and the form of the optimal solution, if one exists?

This work aims to address all of these questions under the restricted setting of constant inputs and no noise.

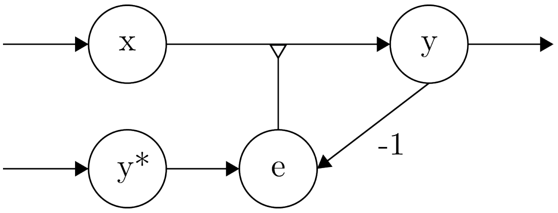

Figure 1: Network diagram used to analyze the PES rule. A connection from x to y learns to output y^* by minimizing $|y^* - y|$. The error signal forms a modulatory connection to modify the weights between x and y.

2 Prescribed Error Sensitivity

For the analysis, we setup a network in Nengo containing an ensemble of n neurons, inject a constant scalar input x, and minimize an error signal $e = y^* - y$ using the PES rule, where y is the scalar output of the ensemble, and y^* is the ideal output (see Fig. 1). The connection weights then learn to represent y^* when presented with x. This work naturally extends to the case where x is a vector, and so we only consider the scalar case for simplicity.

Let $\mathbf{a} \in \mathbb{R}^n$ be the rate activity of each neuron without noise. This is given by the first principle of the NEF, and depends only on x. From here on we assume that $\mathbf{a} \neq \mathbf{0}$, otherwise our ensemble is not properly encoding x, in which case PES will have no effect. The PES rule then applies the following update rule at each timestep:

$$\Delta \mathbf{d} = \kappa e \mathbf{a} \tag{1}$$

where $\kappa > 0$ is the learning rate, and $\mathbf{d} \in \mathbb{R}^n$ is the decoding vector for the connection weight matrix such that, by the first principle of the NEF,

$$y = \mathbf{d}^T \mathbf{a}. \tag{2}$$

Now, to formulate the problem, let $\mathbf{d}[k] \in \mathbb{R}^n$ be the value of the decoding vector $\mathbf{d}$ at the k^{th} timestep of the Nengo simulation, for all $k \in \mathbb{N}$. Our goal is to give a closed-form expression for $\mathbf{d}[k]$ and its convergent solution. For notational convenience, the use of ∞ will imply a limit (e.g. $\mathbf{d}[\infty] := \lim_{k\to\infty} \mathbf{d}[k]$). Therefore, we will answer the following: given κ, $\mathbf{a}$, y^*, and $\mathbf{d}[0]$, what is $\mathbf{d}[k]$ for $k \in \mathbb{N}$ and $\mathbf{d}[\infty]$?

3 Theorem

Let $\gamma = (1 - \kappa \|\mathbf{a}\|^2) < 1$, and $e_0 = y^* - \mathbf{d}[0]^T \mathbf{a}$ be the initial error, then

$$\mathbf{d}[k] = \mathbf{d}[0] + e_0 \frac{\mathbf{a}}{\|\mathbf{a}\|^2}(1 - \gamma^k), \quad k \in \mathbb{N} \tag{3}$$

and so the error at timestep k is $y^* - \mathbf{d}[k]^T \mathbf{a} = e_0 \gamma^k$. In particular, if $\gamma > -1$, then $\mathbf{d}[\infty]$ exists and is given by the unique solution,

$$\mathbf{d}[\infty] = \mathbf{d}[0] + e_0 \frac{\mathbf{a}}{\|\mathbf{a}\|^2}. \tag{4}$$

On the other hand, if $\gamma \leq -1$ then the system is unstable.

3.1 Discussion

An immediate consequence is that the error $y^* - y$ goes to 0 exponentially with rate γ. When $\gamma < 0$, this results in oscillations around the optimal solution. When $\gamma > -1$, the limit exists and is given by the stable optimum (4).

The error given by (3) implies that if we desire an approximation factor of ϵ times the initial error in k timesteps, then set the learning rate to

$$\kappa = (1 - \epsilon^{1/k})/\|a\|^2.$$

The theorem essentially states that the optimal solution is achieved by increasing the decoders by the normalized activity vector scaled by the initial error. In hindsight, this should be somewhat intuitive. If we look at (1), only a scaled version of $\mathbf{a}$ is ever added to the decoders, and so it makes sense that the final difference could only be some multiple of $\mathbf{a}$. If we suppose that $\mathbf{d}[\infty]$ converges, and furthermore $\mathbf{d}[\infty]^T\mathbf{a} = y^*$, then we may indeed solve $\mathbf{d}[\infty]^T\mathbf{a} = \mathbf{d}[0]^T\mathbf{a} + \alpha\mathbf{a}^T\mathbf{a}$ for the required scaling factor $\alpha = \frac{e_0}{\mathbf{a}^T\mathbf{a}}$. Conversely, we can verify that indeed the above $\mathbf{d}[\infty]$ decodes y^* by evaluating $\mathbf{d}[\infty]^T\mathbf{a} = \mathbf{d}[0]^T\mathbf{a} + e_0\frac{\mathbf{a}^T\mathbf{a}}{\mathbf{a}^T\mathbf{a}} = y^*$.

However, this is not a proof, since we have supposed that the optimal solution exists and is unique, while there are infinitely many solutions to $\mathbf{u}^T\mathbf{a} = y^*$. Specifically, $\mathbf{u} = \mathbf{d}[\infty] + \mathbf{v}$, where $\mathbf{v}^T\mathbf{a} = 0$, all of which are stable since they have zero error. We must therefore show that only one of these solutions is reachable (namely $\mathbf{v} = \mathbf{0}$), and that γ gives the exponential rate of convergence.

4 Proof

The first step is to formulate the equation for $\mathbf{d}[k]$ using linear algebra. Let $k \in \mathbb{N}$. Combining (1) and (2) with $e = y^* - y$, gives us:

$$\begin{aligned}\mathbf{d}[k+1]^T &= \mathbf{d}[k]^T + \kappa(y^* - \mathbf{d}[k]^T\mathbf{a})\mathbf{a}^T \\ &= \mathbf{d}[k]^T\underbrace{(I - \kappa\mathbf{a}\mathbf{a}^T)}_{A} + \underbrace{\kappa y^*\mathbf{a}^T}_{\mathbf{c}^T}.\end{aligned}$$

The last two components are labeled by the matrix $A \in \mathbb{R}^{n,n}$ and the vector $\mathbf{c} \in \mathbb{R}^n$, since they will appear frequently throughout the analysis. This also allows us to express the above relationship compactly as:

$$\mathbf{d}[k+1]^T = \mathbf{d}[k]^T A + \mathbf{c}^T. \tag{5}$$

It is important to note that neither A nor $\mathbf{c}$ depend on $\mathbf{d}[k]$. A depends solely on $\mathbf{a}$, which is fixed for a given x by the first principle of the NEF. $\mathbf{c}$ depends only on $\mathbf{a}$ and y^*, which are again also fixed. Therefore, $\mathbf{d}[k]$, and thus $\mathbf{d}[\infty]$, can be found inductively:

$$\mathbf{d}[k]^T = \mathbf{d}[0]^T A^k + \sum_{i=0}^{k-1}\mathbf{c}^T A^i. \tag{6}$$

In order to find a closed-form expression for (6) we must essentially characterize the effect of repeated multiplication and addition by using the eigendecomposition of the system. This is done by interpreting (5) as the *state-space* *representation* of a discrete linear control system with constant input, where the "state" is given by the current decoders. Since A is symmetric, this is also equivalent to $\mathbf{d}[k+1] = A\mathbf{d}[k] + \mathbf{c}$, but it turns out that the eigendecomposition of the former system has a nicer interpretation.

We proceed with some basic results on $A = I - \kappa \mathbf{a}\mathbf{a}^T$, and then move toward a construction that reveals the eigenvectors of the whole system, which in turn proves the main theorem.

Since A is symmetric, by the spectral theorem it can be diagonalized to $A = VWV^T$, where V is orthonormal ($V^{-1} = V^T$), and W is diagonal. Moreover, the columns of V are the unit eigenvectors of A (giving us an n-dimensional eigenbasis), and their corresponding eigenvalues are on the respective diagonals in W. Now, since A differs only from I by an outer product, it has very specific structure.

Lemma 4.1. $\mathbf{a}$ *is an eigenvector of* A *with eigenvalue* $\gamma < 1$.

Proof. Recalling that $\gamma = 1 - \kappa\|\mathbf{a}\|^2$,

$$\begin{aligned} A\mathbf{a} &= (I - \kappa \mathbf{a}\mathbf{a}^T)\mathbf{a} \\ &= (\mathbf{a} - \mathbf{a}\kappa\mathbf{a}^T\mathbf{a}) \\ &= \gamma\mathbf{a} \end{aligned}$$

Since $\mathbf{a} \neq 0$ is non-negative, we know $\|\mathbf{a}\|^2 > 0$ (nd $\kappa > 0$), thus $\gamma < 1$. □

Lemma 4.2. *The remaining* $n-1$ *eigenvectors of* A *are orthogonal to* $\mathbf{a}$ *with eigenvalue 1.*

Proof. The first part follows trivially from the fact that A can be diagonalized and so the remaining $n-1$ dimensions must span a subspace that is orthogonal to $\mathbf{a}$ (by lemma 4.1). To be precise, the remaining eigenvectors form an orthonormal basis for the nullspace of $\mathbf{a}^T$, with eigenvalue 1, because $A\mathbf{u} = \mathbf{u} \iff (I - \mathbf{a}\mathbf{a}^T)\mathbf{u} = \mathbf{u} \iff \mathbf{0} = \mathbf{a}\mathbf{a}^T\mathbf{u} \iff \mathbf{a}^T\mathbf{u} = 0$, since $\mathbf{a} \neq 0$. □

Geometrically, this means that the mapping A modifies a single dimension of the vector, namely the line spanning $\mathbf{a}$ with eigenvalue $\gamma < 1$. The remaining portion, in the hyperplane orthogonal to $\mathbf{a}$, remains untouched. Given conditions on κ so that $\gamma > -1$, in the limit A^∞ will zero out the dimension spanned by $\mathbf{a}$. In fact, these two lemmas are enough to diagonalize A, which we use to show that $A^\infty = I - \frac{\mathbf{a}\mathbf{a}^T}{\|\mathbf{a}\|^2}$ (below). This makes intuitive sense since $P_\mathbf{a} = \frac{\mathbf{a}\mathbf{a}^T}{\|\mathbf{a}\|^2}$ is a projection onto $\mathbf{a}$, and so $A^\infty\mathbf{u} = \mathbf{u} - P_\mathbf{a}\mathbf{u}$ gives us the portion of $\mathbf{u}$ that is orthogonal to $\mathbf{a}$. To see this more rigorously, let V_0 be equal to V with all but its first column ($\frac{\mathbf{a}}{\|\mathbf{a}\|}$) set to zero. That is, $V = V_0 + \begin{pmatrix} 0 & V_1 \end{pmatrix}$. Then since the first diagonal of W^∞ goes to zero,

$$\begin{aligned} A^\infty &= VW^\infty V^T \\ &= (V - V_0)V^T \\ &= I - V_0V^T \\ &= I - \frac{\mathbf{a}}{\|\mathbf{a}\|}\frac{\mathbf{a}^T}{\|\mathbf{a}\|} \\ &= I - \frac{\mathbf{a}\mathbf{a}^T}{\|\mathbf{a}\|^2}. \end{aligned}$$

Of course, this is only half the story, since we have not yet involved $\mathbf{c}$, which is the sole bearer of y^*. To this end, we use the following construction:

$$\tilde{A} = \begin{pmatrix} A & \mathbf{0} \\ \mathbf{c}^T & 1 \end{pmatrix} \in \mathbb{R}^{n+1,n+1}, \quad \tilde{\mathbf{d}}[k] = \begin{pmatrix} \mathbf{d}[k] \\ 1 \end{pmatrix} \in \mathbb{R}^{n+1}.$$

It is easy to see that $\tilde{\mathbf{d}}^T[k]\tilde{A}$ will multiply $\mathbf{d}[k]$ by A and add $\mathbf{c}^T$, so by induction this allows us to rewrite (5) as a single matrix multiply:

$$\tilde{\mathbf{d}}[k]^T = \tilde{\mathbf{d}}[0]^T \tilde{A}^k. \tag{7}$$

Lemma 4.3. *Let* $\beta = \frac{\mathbf{c}^T\mathbf{a}}{(\gamma-1)\|\mathbf{a}\|}$*, then* $\tilde{\mathbf{a}} = \begin{pmatrix} \frac{\mathbf{a}}{\|\mathbf{a}\|} \\ \beta \end{pmatrix}$ *is an eigenvector of* $\tilde{A}$ *with eigenvalue* γ*.*

Proof. $\tilde{\mathbf{a}}$ is well-defined since $\gamma < 1$.
$\tilde{A}\tilde{\mathbf{a}} = \begin{pmatrix} \frac{A\mathbf{a}}{\|\mathbf{a}\|} \\ \mathbf{c}^T\frac{\mathbf{a}}{\|\mathbf{a}\|} + \beta \end{pmatrix} = \begin{pmatrix} \frac{\gamma\mathbf{a}}{\|\mathbf{a}\|} \\ \frac{\mathbf{c}^T\mathbf{a}\gamma}{(\gamma-1)\|\mathbf{a}\|} \end{pmatrix} = \gamma\tilde{\mathbf{a}}$, by lemma 4.1. □

We remark that the last component of $\tilde{\mathbf{a}}$ can be simplified to:

$$\beta = \frac{\mathbf{c}^T\mathbf{a}}{(\gamma-1)\|\mathbf{a}\|} = \frac{\kappa y^*\mathbf{a}^T\mathbf{a}}{-\kappa\|\mathbf{a}\|^2\|\mathbf{a}\|} = -\frac{y^*}{\|\mathbf{a}\|}.$$

Lemma 4.4. *Suppose* $\mathbf{u}$ *is an eigenvector of* A *such that* $A\mathbf{u} = \mathbf{u}$*, then* $\tilde{\mathbf{u}} = \begin{pmatrix} \mathbf{u} \\ 0 \end{pmatrix}$ *similarly satisfies* $\tilde{A}\tilde{\mathbf{u}} = \tilde{\mathbf{u}}$*.*

Proof.

$$\tilde{A}\tilde{\mathbf{u}} = \begin{pmatrix} A\mathbf{u} \\ \mathbf{c}^T\mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{u} \\ \kappa y^*\mathbf{a}^T\mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{u} \\ 0 \end{pmatrix} = \tilde{\mathbf{u}}$$

since $\mathbf{a}^T\mathbf{u} = 0$ from lemma 4.2. □

Now observe that $(0, 0, \ldots, 1)^T \in \mathbb{R}^{n+1}$ is an eigenvector for $\tilde{A}$ with eigenvalue 1, and so we have found $n+1$ eigenvectors for $\tilde{A}$. Recall that $A = VWV^T$. Without loss of generality, suppose the columns of V are arranged so that $\frac{\mathbf{a}}{\|\mathbf{a}\|}$ is the first column (using lemma 4.1). Then, construct $\mathbf{b} = (\beta, 0, \ldots, 0)^T \in \mathbb{R}^{n+1}$, and place all $n+1$ eigenvectors in a square matrix,

$$\tilde{V} = \begin{pmatrix} V & \mathbf{0} \\ \mathbf{b}^T & 1 \end{pmatrix}$$

with corresponding eigenvalues

$$\tilde{W} = \begin{pmatrix} W & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

Then $\tilde{A}\tilde{V} = \tilde{V}\tilde{W}$ by lemmas 4.3 and 4.4. To complete the diagonalization, we must show that $\tilde{V}$ is invertible by arguing that the eigenvectors we found are linearly independent, or better yet by explicitly constructing the inverse:

$$\tilde{V}^{-1} = \begin{pmatrix} V^T & \mathbf{0} \\ -\mathbf{b}^T V^T & 1 \end{pmatrix} = \begin{pmatrix} V^T & \mathbf{0} \\ -\beta\frac{\mathbf{a}^T}{\|\mathbf{a}\|} & 1 \end{pmatrix}.$$

Finally, we are ready to diagonalize $\tilde{A}$:

Lemma 4.5. $\tilde{A} = \tilde{V}\tilde{W}\tilde{V}^{-1} = \begin{pmatrix} V & \mathbf{0} \\ \mathbf{b}^T & 1 \end{pmatrix} \begin{pmatrix} W & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} V^T & \mathbf{0} \\ -\mathbf{b}^T V^T & 1 \end{pmatrix}$.

Proof.

$$\tilde{V}\tilde{V}^{-1} = \begin{pmatrix} V & \mathbf{0} \\ \mathbf{b}^T & 1 \end{pmatrix} \begin{pmatrix} V^T & \mathbf{0} \\ -\mathbf{b}^T V^T & 1 \end{pmatrix} = \begin{pmatrix} VV^T & \mathbf{0} \\ \mathbf{b}^T V^T - \mathbf{b}^T V^T & 1 \end{pmatrix} = I$$

□

We are now ready to prove the main theorem. The purpose of obtaining the above diagonalization is that it enables us to give a closed-form solution for $\tilde{\mathbf{d}}[k]$ via (7) and lemma 4.5. Let $V_1 \in \mathbb{R}^{n,n-1}$ be equal to V without its first column, such that $V = \begin{pmatrix} \frac{\mathbf{a}}{\|\mathbf{a}\|} & V_1 \end{pmatrix}$.

$$\begin{aligned}
\mathbf{d}[k]^T &= \begin{pmatrix} \mathbf{d}[0] \\ 1 \end{pmatrix}^T \begin{pmatrix} \frac{\mathbf{a}}{\|\mathbf{a}\|} & V_1 & 0 \\ \beta & \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \gamma^k & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \begin{pmatrix} \frac{\mathbf{a}^T}{\|\mathbf{a}\|} \\ V_1^T \\ -\beta \frac{\mathbf{a}^T}{\|\mathbf{a}\|} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{d}[0]^T \frac{\mathbf{a}}{\|\mathbf{a}\|} + \beta & \mathbf{d}[0]^T V_1 & 1 \end{pmatrix} \begin{pmatrix} \gamma^k & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \begin{pmatrix} \frac{\mathbf{a}^T}{\|\mathbf{a}\|} \\ V_1^T \\ -\beta \frac{\mathbf{a}^T}{\|\mathbf{a}\|} \end{pmatrix} \\
&= \begin{pmatrix} \gamma^k(\mathbf{d}[0]^T \frac{\mathbf{a}}{\|\mathbf{a}\|} + \beta) & \mathbf{d}[0]^T V_1 & 1 \end{pmatrix} \begin{pmatrix} \frac{\mathbf{a}^T}{\|\mathbf{a}\|} \\ V_1^T \\ -\beta \frac{\mathbf{a}^T}{\|\mathbf{a}\|} \end{pmatrix} \\
&= \gamma^k(\mathbf{d}[0]^T \frac{\mathbf{a}}{\|\mathbf{a}\|} + \beta) \frac{\mathbf{a}^T}{\|\mathbf{a}\|} + \mathbf{d}[0]^T V_1 V_1^T - \beta \frac{\mathbf{a}^T}{\|\mathbf{a}\|} \\
&= \mathbf{d}[0]^T (I - \frac{\mathbf{a}\mathbf{a}^T}{\|\mathbf{a}\|^2}) + \frac{y^*}{\|\mathbf{a}\|^2}\mathbf{a}^T - \gamma^k (y^* - \mathbf{d}[0]^T \mathbf{a}) \frac{\mathbf{a}^T}{\|\mathbf{a}\|^2} \\
&= \mathbf{d}[0]^T + e_0 \frac{\mathbf{a}^T}{\|\mathbf{a}\|^2}(1 - \gamma^k)
\end{aligned}$$

To find the limit as $k \to \infty$, recall from lemmas 4.3 and 4.4 that the spectrum of the system is $sp(\tilde{A}) = sp(A) = \{\gamma, 1, 1, ...\}$, by lemma 4.1. Thus, (7) requires that $|\gamma| < 1$ in order for $\mathbf{d}[\infty]$ to exist. Since $\gamma < 1$, $\mathbf{d}[\infty]$ converges $\iff$ $\gamma > -1$. If $\gamma \leq -1$, then the state explodes. In the stable condition,

$$\mathbf{d}[\infty]^T = \mathbf{d}[0]^T \underbrace{(I - \frac{\mathbf{a}\mathbf{a}^T}{\|\mathbf{a}\|^2})}_{A^\infty} + \underbrace{\frac{y^*}{\|\mathbf{a}\|^2}\mathbf{a}^T}_{\sum_{i=0}^{\infty} \mathbf{c}^T A^i}$$

To get the error at timestep k, we use (2) to obtain

$$(\mathbf{d}[\infty]^T - \mathbf{d}[k]^T)\mathbf{a} = e_0 \gamma^k \frac{\mathbf{a}^T \mathbf{a}}{\|\mathbf{a}\|^2} = e_0 \gamma^k$$

which concludes the proof. □

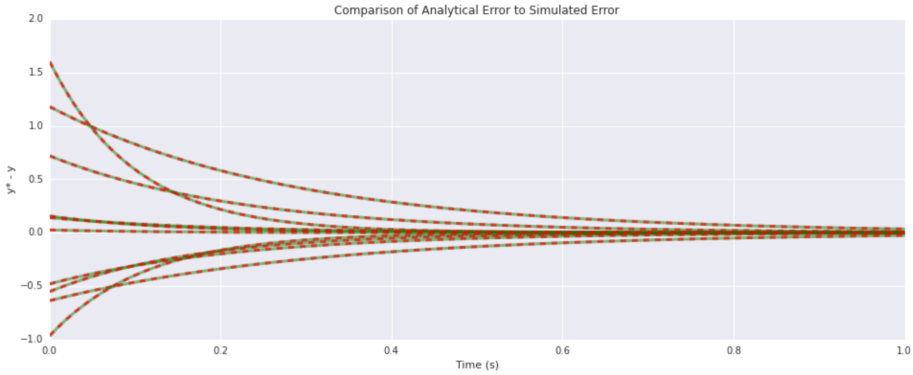

Figure 2: Comparison of the analytical error given by the theorem to the simulated error in Nengo. Each line indicates the error $y^* - y$ over time for a single trial, for a random number of rate neurons (noiseless) from the range $[10, 200]$, a random learning rate from $[10^{-6}, 5 \times 10^{-4}]$, a random x from $[-1, 1]$, and a random y^* from $[-1, 1]$. In all trials, the analytical error is shown with a dashed line, with a root-mean-square error of less than 10^{-14}.

References

Trevor Bekolay, Carter Kolbeck, and Chris Eliasmith. Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks. In *35th Annual Conference of the Cognitive Science Society*, pages 169–174. Cognitive Science Society, 2013.

Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Russell Voelker, and Chris Eliasmith. Nengo: A python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7(48), 2014. ISSN 1662-5196. doi: 10.3389/fninf.2013.00048.

Chris Eliasmith and Charles H. Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems.* MIT Press, Cambridge, MA, 2003.

Brent Komer. Biologically inspired adaptive control of quadcopter flight. Masters thesis, University of Waterloo, Waterloo, ON, 2015. URL `https://uwspace.uwaterloo.ca/handle/10012/9549`.

Daniel Rasmussen and Chris Eliasmith. A neural model of hierarchical reinforcement learning. In Paul Bello, Marcello Guarini, Marjorie McShane, and Brian Scassellati, editors, *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 1252–1257, Austin, 2014. Cognitive Science Society. URL `https://mindmodeling.org/cogsci2014/papers/221/index.html`.

Oliver Trujillo and Chris Eliasmith. A spiking-neuron model of memory encoding and replay in hippocampus. In *BMC Neuroscience*, page 166. Organization for Computational Neurosciences, 2014.