Towards adaptive spiking label propagation

Kathleen E. Hamilton
Catherine D. Schuman

July, 2018

International Conference on Neuromorphic Systems (ICONS 2018)

https://ornlcda.github.io/icons2018/

Copyright ACM. Please find the ACM version of the final camera-ready paper at http://dl.acm.org/citation.cfm?id=3229884

The online home for this paper may be found at: http://neuromorphic.eecs.utk.edu

Citation – Plain Text:

```
author          K. E. Hamilton and C. D. Schuman
title           Towards Adaptive Spiking Label Propagation
booktitle       International Conference on Neuromorphic Computing Systems
publisher       ACM
address         Knoxville, TN
month           July
year            2018
```

Bibtex:

```
@INPROCEEDINGS{hs:18:tas,
    author = "K. E. Hamilton and C. D. Schuman",
    title = "Towards Adaptive Spiking Label Propagation",
    booktitle = "International Conference on Neuromorphic Computing Systems",
    publisher = "ACM",
    address = "Knoxville, TN",
    month = "July",
    year = "2018"
}
```
Towards adaptive spiking label propagation

Kathleen E. Hamilton
Oak Ridge National Library
Oak Ridge, Tennessee
hamiltonke@ornl.gov

Catherine D. Schuman
Oak Ridge National Library
Oak Ridge, Tennessee
schumancd@ornl.gov

ABSTRACT
Graph algorithms are a new class of applications for neuromorphic hardware. Rather than adapting deep learning and standard neural network approaches to a low-precision spiking environment, we look at how graph algorithms can be redesigned to incorporate and extract information generated by spiking neurons. While fully connected spin glass implementations of spiking label propagation have shown promising results on graphs with dense communities, identifying sparse communities remains difficult. This work focuses on steps towards an adaptive spike-based implementation of label propagation, utilizing sparse embeddings and synaptic plasticity. Sparser embeddings reduce the number of inhibitory connections and synaptic plasticity is used to simultaneously amplify spike responses between neurons in the same community, while impeding spike responses across different communities. We present results on identifying communities in sparse graphs, focusing on graphs with very sparse communities.

CCS CONCEPTS
- Mathematics of computing → Graph algorithms; Random graphs;
- Hardware → Emerging tools and methodologies; Neural systems;

KEYWORDS
neuromorphic, community detection, graph algorithm, path finding, spiking neural networks

ACM Reference Format:

1 INTRODUCTION
Neuromorphic computing is commonly referred to as "brain-inspired" computing, utilizing hardware that executes processes similar to how the mammalian brain executes cognitive function. These systems have predominantly been used to accelerate deep learning algorithms [5, 13] but there has been growing interest in adapting non-deep learning algorithms and applications to these systems. For example, a spike-based implementation of simulated annealing has been used to solve various constraint satisfaction problems on the SpiNNaker system [3, 11].

Neuromorphic hardware is developed to mimic the biological processes of the mammalian brain, and one important element is synaptic plasticity. Recent hardware developments have focused on building neuromorphic systems that can update and change the weights associated with individual synapses (e.g. Intel’s Loihi chip [2]). In this work we present an adaptation of spiking label propagation [6, 7], which was previously implemented with static synapses and fully connected spiking neural networks (SNNs). Our adapted algorithm is deployed on larger, sparser networks and utilizes plastic synapses and a sparse embedding of a network into a spiking neuron system.

The previous studies of spiking label propagation [6, 7, 14, 15] have demonstrated that communities can be identified on graphs from firing output. In [14, 15] the use of global driving and global control lead to synchronization over large populations of neurons, which was used to infer graph communities. In [6, 7], the use of local driving and localized spike responses were used to identify communities, based on similarities in the local spike responses.

The label propagation method introduced in [7] was inspired by spin glass physics, where local interactions can lead to localized patterns without influencing global behavior. Similar approaches for spin-glass inspired community detection, implemented with interacting spins [9, 16], have been shown to avoid the resolution limit which is common in other modality-based measures [1]. However, when implemented with spiking neurons and homogeneous, static synapses, it was seen in [6] that the method has difficulty identifying small communities in sparse graphs.

We modify spiking label propagation to address two challenges: generating a local spike response that can sufficiently cover a sparse community, and efficiently embedding a graph in a spiking neuron system with minimal synaptic connections. We use synaptic connections that are now able to implement plasticity-based learning rules, and the embedding of a graph into a SNN is made sparser by reducing the number of inhibitory synapses. The goal of these modifications is to allow for the localized spike response to grow larger as more neurons in a community are driven without causing spiking cascades that spread throughout the network.

These modifications are designed to overcome challenges faced when analyzing real world networks. Real world networks can
our custom spiking neural network simulation software frame-
work is written in C++ and utilizes a discrete-event simulation
approach for rapid network evaluation. The simulator framework
can be run in either serialized or parallel mode and can scale to
utilize a distributed memory supercomputer or cluster in order to
simulate the activity in larger networks (>1000 neurons with dense
connectivity) more rapidly than on a single processor [17]. The
software framework is modular and allows for different neuron
models and synapse models to be easily implemented and evaluated
using the simulation engine.

The spiking dynamics are simulated with the following neuron
parameters: \( v_{th} = 0.8 \text{ V}, v_0 = v_R = 0.0 \text{ V}, \tau = 25 \text{ ms}, \tau_g = 20 \text{ ms}, \)
and synaptic weight amplitude \( |w_{ij}| = 0.75\text{ V}. \) Driving is done by
sending 10 spikes to a neuron, spaced at 0.21 ms intervals.

Once the full spike raster was generated, the label propagation
algorithm of [6] was applied. Each spike train was decoded into
a binary vector \( x_i \) using a time bin width of \( \Delta t = 0.03 \text{ sec} \), and the
Hamming metric \( H(x_i, x_j) \) defined in [10] was used to quantify
the degree of similarity between pairs of decoded spike trains. A
threshold value \( h_0 \) for the Hamming similarity was chosen and fixed,
and each neuron of \( S \) was initialized with a unique label. Labels
propagate through the system based on the degree of similarity
between pairs of neuron spike trains; if \( H(x_i, x_j) \geq h_0 \) then the
label of \( x_i \) is applied to \( x_j \).

With each benchmark graph in Table 1, we also have a set of
known labels which act as the ground truth. This known distribution
of labels over the original vertex set \( \{v_i\} \) is compared to the label
propagation output, which returns a predicted label distribution

| Graph | \( |V(G)| \) | \( |E(G)| \) | \( |\{Q_i\}| \) | \( D \) | \( \sigma \) |
|-------|-----------|-----------|-------------|-----|--------|
| \( G_0 \) | 128       | 256       | 16          | 4   | 0.032  |
| \( G_1 \) | 128       | 1024      | 4           | 16  | 0.126  |
| \( G_2 \) | 256       | 1014      | 16          | 7.9 | 0.031  |

Figure 1: The benchmark graph on 128 vertices with 16
unique communities. Many communities can be discon-
nected from the graph by removing 2 edges, but to discon-
nect a single vertex from a community requires the removal of
>2 edges.
Figure 2: The comparison between binary decoded spikes trains generated under the fully connected spin glass model. Individual communities are not significantly different and similarity within individual communities is not significantly high.

(L'). The difference between the two distributions is quantified using the variance of information $\text{VI}(L_0, L')$ [12]. When $\text{VI}(L_0, L') = 0$ then the two distributions are identical.

3 SNN CONSTRUCTION

In the following sections we show how the spiking response of a system, and the ability to recover a known label distribution, is affected by synapse plasticity, and the sparsity of the SNN.

3.1 Sparse embedding of local spin glass networks

Fully connected spin glass models have been used to demonstrate proof-of-concept results that spiking neuron systems can be used to implement label propagation. However fully connected systems will be dominated by inhibitory synaptic connections for most real-world networks, which tend to be sparse. The embedding discussed in this section is used for spiking label propagation, where the generation of a localized response from a driving a single neuron is needed. For the sparse graphs we consider in this work, embedding into a fully connected spin glass will result in a graph dominated by inhibitory synapses. This leads to very sparse spike rasters and as seen in Fig. 2 loss of distinction between small communities. We now define a spiking neuron system which has excitatory synapses for any edge on the original graph, but inhibitory synapses are only added between small localized regions on the graph, identified from the graph adjacency matrix.

The design of our spiking neuron systems for label propagation require that two spikes must arrive from a driven neuron in order to cause a neuron to fire in response. A neuron can fire when it is directly connected to a driven neuron, or if there are a number of short paths (length 2 paths) connecting to the driven neuron. Inhibitory connections are only added if there are $a_0$ length 2 paths between two neurons, and neurons that are far apart on the graph are left disconnected. On a sparse graph, there may be few neurons connected by multiple length-2 paths, as a result the choice of $a_0$ can lead to a SNN with very few inhibitory connections. The number of inhibitory connections added to the SNN are given in Table 2.

\begin{equation}
\omega_{ij} < 0 \in W(S) \iff (A^2)_{ij} = a_0
\end{equation}

In Table 2 we compare the number of inhibitory, and excitatory connections for various values of $a_0$. A set of sparse embeddings were generated for $G_0$, and the spiking dynamics were generated using only static synapses. Reducing the number of inhibitory synapses in the SNN results in a larger spike response from driving a single neuron. This can be quantified by simply counting the number of neurons that are active when one neuron is driven (see Fig. 3). In Fig. 4 the sparse embeddings on $G_0$ lead to an increased degree of dissimilarity between different communities when $a_0 = 1$ or $a_0 = 3$. Label propagation is run on the decoded spike output for each sparse embedding shows that for $G_0$. The minimal variance of information is obtained for the sparser embeddings with $a_0 = 1$ and $a_0 = 3$.

3.2 Synapse plasticity

We implement spike timing dependent plasticity using Hebbian learning. Only one neuron fires spikes due to driving by an external source, while any other neuron that fires during the active driving of a neuron will fire due to “internal firing,” firing a spike because of the arrival of multiple spikes from a neighboring neurons. We
Figure 4: The degree of similarity between binary decoded spike trains for a benchmark graph with 16 communities each with 8 vertices using the adjacency defined embedding and static synapses. (a) shows the similarity between spike trains with inhibitory connections added only to pairs of disconnected neurons which have 1 length-2 paths between them ($a_0 = 1$), (b) shows the similarity between spike trains generated under the sparser embedding, with inhibitory connections added only to pairs of disconnected neurons which have 2 length-2 paths between them ($a_0 = 2$), (c) shows the similarity between spike trains generated under the sparser embedding, with inhibitory connections added only to pairs of disconnected neurons which have 3 length-2 paths between them ($a_0 = 3$). Connections weighted according to Eq. 4.

Figure 5: The number of unique labels at the end of label propagation (solid lines) and the resulting variance of information (dashed lines) for benchmark graph $G_0$. The full spin-glass embedding is compared to results generated from each of the sparse embeddings used to generate Fig. 4.

We define the learning rule according to the total output of the neurons $n_i, n_j$ during the time that neuron $n_j$ is driven by an external source $\Delta(t_j)$.

We implement plasticity and synaptic learning through a simple update rule: after the external driving of a neuron ($n_i$) ends, a subset of synaptic weights are updated based upon their spiking behavior during the external driving. Only synapses which terminate at $n_i$ (have $n_i$ as a post-synaptic neuron) will update: $\{ w_{ij} \} \in W(S)$. We define $\eta(n_j)$ over the entire period of external driving $\Delta_{EXT}$: $\eta(n_j) = 1$ if $n_j$ fires a spike during $\Delta_{EXT}$ and $\eta(n_j) = 0$ if $n_j$ does not fire at all during $\Delta_{EXT}$.

If $n_j$ does not fire, then $s'_{w} < s_w$. If $n_j$ does fire, then $s'_{w} > s_w$. We used two sets of learning rates: in Sec. 3.2.1 the learning rate was homogeneous and chosen to be $\Delta_{k} = |s_w|$ for all synapses, in Sec. 3.2.2 the learning rate was dependent on the local connectivity of the underlying graph.

3.2.1 Homogeneous plastic synapses. If $n_j$ fires in response to $n_i$ being driven, then the excitatory synapse $s_{ji}$ is increased to $2|s_w|$, which exceeds the spiking threshold $2|s_w| > v_{th}$. This ensures that when $n_j$ is driven, the strengthened synapse is able to immediately cause response spiking in $n_i$. While excitatory synapses are increased, inhibitory synapses are further depressed: if $n_j$ does not fire when $n_i$. This is vitally important to prevent runaway spike cascades from spreading through the entire system. Additionally, there are inhibitory synapses that are reduced in magnitude, but do not become excitatory: for neurons that are densely connected (e.g. a large number of length-2 paths exist between $n_i$ and $n_j$) but not directly connected, the fact that $n_j$ will still spike when $n_i$ is driven will reduce the strength of the inhibitory synapse $s_{ji}$ and depending on the value of $s_{ji}$ this synapse may be effectively removed ($s_{ji} = 0$).

Incorporating synapse plasticity leads to a system with dynamic spiking behavior: the local spike response of a single neuron is not just determined by the local connectivity but is also dependent on whether or not a neighboring neuron has been driven recently. The benchmark graph shown in Fig. 1, has 128 vertices, 256 edges and is degree regular with all vertices having degree $k = 4$. When it is embedded into a spiking neuron system as a fully connected
spin glass, this translates to a system with 256 symmetric pairs of excitatory synapses, and 7872 symmetric pairs of inhibitory synapses. This leads to extremely localized spike responses when driving individual neurons. Introducing plasticity to the synapses will lead to more varied spike responses, but it can lead to runaway spike cascades (see Fig. 6).

### 3.2.2 Heterogeneous plastic synapses

The final network design we considered combined multiple sparse embedding with plastic synapses, where now the initial weight of the inhibitory synapses was dependent on the local connectivity of the original graph, and the synapses are plastic. Rather than choose a single value of $a_0$, we use $a_0 = 1, 2, 3$ to add inhibitory synapses of different initial weights.

\begin{align*}
    w_{ij} \in W(S), \quad s_{w} &= -0.75 \text{ V} \iff (A^2)_{ij} = 1 \\
    w_{ij} \in W(S), \quad s_{w} &= -0.5 \text{ V} \iff (A^2)_{ij} = 2 \\
    w_{ij} \in W(S), \quad s_{w} &= -0.25 \text{ V} \iff (A^2)_{ij} = 3
\end{align*}

Thus the strongest inhibitory weights are added between neurons with the sparsest connections. For this embedding, the learning rate was reduced to $\Delta_s = 0.25 \text{ V}$. This learning rate is large enough to ensure that excitatory synapses increase to a weight that surpasses the spike threshold.

The heterogeneous SNN has dynamic spiking behavior, but does not lead to cascades. In Fig. 7 we show a set of embeddings that have a bounded number of spiking neurons during external driving for static synapses, homogeneous plastic synapses, and heterogeneous synapses. In Fig. 8 we compare the final label sets returned using the original iteration of label propagation (with static synapses embedded into a full spin-glass), the sparse embedding ($a_0 = 1$) with plasticity, and the system with sparse embedding and heterogeneous starting weights.

### 4 DISCUSSION

Previous implementations of spiking label propagation used fully connected spin glass systems with static and homogeneous synaptic weights \([6, 8]\). While these systems could generate spiking responses that could accurately identify large, densely-connected communities, locating and isolating small, sparse communities is difficult. The results in Secs. 3.1, 3.2.1 and 3.2.2 indicate that performance on sparse graphs with small communities can be improved by implementing synapse plasticity and sparse embeddings. We compare these results to a graph on which the original iteration of label propagation performed very well, the standard Girvan-Newman benchmark.
We have modified the SNNs used for spiking label propagation. For all graphs, the embedding returned improvements to label propagation, even with static synapses. The variance of information for benchmark graph \( G_0 \) led to improved performance on homogeneous plastic synapses, and the sparse embedding applied to a dense graph may lead to runaway spike cascades.

Synapse plasticity was introduced, but its usefulness is dictated by the graph. Homogeneous learning rates do not add any significant improvements in performance to systems with uniform synaptic weights. Embeddings that can generate a localized spike response that do not cover a significant fraction of the network (e.g., the full spin glass or the \( a_0 = 0 \) embedding shown in Fig. 10) do not undergo significant changes when plasticity is implemented. However, embeddings that generate large spike responses can quickly be driven into spiking cascades by the addition of plasticity (e.g., the \( a_0 = 2, 3 \) embeddings shown in Fig. 6). If an embedding already generates a spike cascade with static synapses, the addition of plasticity will not mitigate this (see the \( a_0 = 2 \) embedding shown in Fig. 10). On the other hand, initializing the synaptic weights over a heterogeneous set of values, can lead to bounded, localized spike behavior.

The optimal embedding for a graph is determined by the local connectivity of the graph. The results we have presented in this work are the first steps towards a fully adaptive implementation of spiking label propagation. We continue to work towards a spiking neural system that can learn and adapt its spiking characteristics to amplify local spike responses, and also act to inhibit spike cascades. The sparse embeddings shown in this work show that by reducing the number of inhibitory synapses, the local spike responses can be amplified. Inhibiting spike cascades, or mitigating their spread once they have begun, remains an open question.

Acknowledgments

Research sponsored in part by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

References

Figure 9: The degree of similarity between binary decoded spike trains for the Girvan-Newman benchmark graph. (a) shows the similarity between spike trains with inhibitory connections added only to pairs of disconnected neurons which have 1 length-2 paths between them \(a_0 = 1\), (b) shows the similarity between spike trains generated under the sparser embedding, with inhibitory connections added only to pairs of disconnected neurons which have 2 length-2 paths between them \(a_0 = 2\), (c) shows the similarity between spike trains generated under the sparser embedding, with inhibitory connections added only to pairs of disconnected neurons which have 3 length-2 paths between them \(a_0 = 3\). connections weighted according to Eq. 4.

Figure 10: The number of neurons that fire when a single neuron is driven. For the full spin-glass and the \(a_0 = 1\) embedding with static synapses (solid black, red) the spike response is bounded, and this localized response is maintained with plastic synapses (dashed black, dashed red). Under sparser embeddings, the number of spiking neurons can cover the entire network, for either static or plastic synapses (blue and green lines).

Figure 11: The number of unique labels at the end of label propagation and the resulting variance of information for benchmark graph \(G_1\). The full spin-glass embedding with static synapses is compared the \(a_0 = 1\) sparse embedding with homogeneous plastic synapses, and the sparse embedding with heterogeneous plastic synapses.


