The Remarkable Robustness of Surrogate Gradient Learning for Instilling Complex Function in Spiking Neural Networks
AI5073

Kartik Srinivas, Susmit, Deepika

March 2023

Introduction

- Deep neural networks - lack cell type diversity and do not obey Dale's law

Introduction

- Deep neural networks - lack cell type diversity and do not obey Dale's law
- Spiking neural networks - more biologically plausible for brain information processing

Introduction

- Deep neural networks - lack cell type diversity and do not obey Dale's law
- Spiking neural networks - more biologically plausible for brain information processing
- Connectivity - functionality relation - not understood well. Impediment to progress.

Introduction

- Deep neural networks - lack cell type diversity and do not obey Dale's law
- Spiking neural networks - more biologically plausible for brain information processing
- Connectivity - functionality relation - not understood well. Impediment to progress.
- Roadblock - Non-differentiable non-linearity of spikes

Introduction

- Deep neural networks - lack cell type diversity and do not obey Dale's law
- Spiking neural networks - more biologically plausible for brain information processing
- Connectivity - functionality relation - not understood well. Impediment to progress.
- Roadblock - Non-differentiable non-linearity of spikes
- Solution - **Use surrogate gradients**

Introduction

- Deep neural networks - lack cell type diversity and do not obey Dale's law
- Spiking neural networks - more biologically plausible for brain information processing
- Connectivity - functionality relation - not understood well. Impediment to progress.
- Roadblock - Non-differentiable non-linearity of spikes
- Solution - **Use surrogate gradients**
- Raises more questions
  - Choice of surrogate
  - Implementation influence on model effectiveness
  - Robustness of surrogates - shape, loss functions, scale, etc

Spiking Neural Networks

- Have a temporal aspect

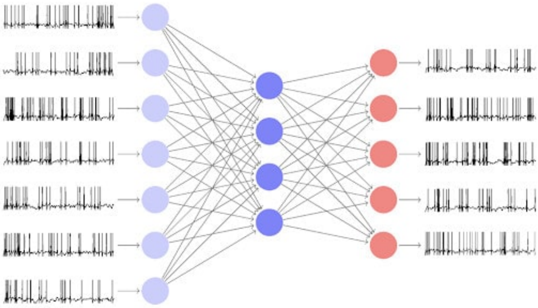


Figure: An SNN

Spiking Neural Networks

- Have a temporal aspect
- Work with spike trains

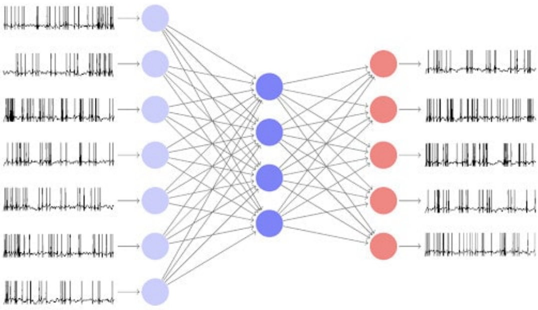


Figure: An SNN

Spiking Neural Networks

- Have a temporal aspect
- Work with spike trains
- Transmit information(fire) only when membrane potential crosses a threshold. Not at each propagation cycle

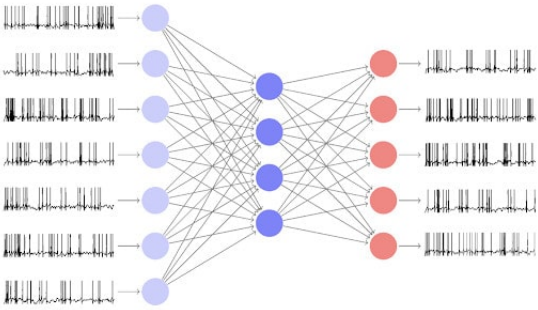


Figure: An SNN

Manifolds

A D - dimensional Manifold is defined by a continuous function f that takes D inputs (all in the range $0 \leq x \leq 1$) and gives a point in M dimensional space as output. (This usually is some curve/sheet in M - dimensional euclidean space) The function used to map the points in the paper is a fourier basis(for each dimension D):-

$$f_i(\vec{x}) = \prod_j \left[\sum_k^{n_{cutoff}} \frac{1}{k^\alpha} \theta^A_{ijk} \sin(2\pi(k x_j \theta^B_{ijk} + \theta^C_{ijk})) \right] \tag{1}$$

Manifold Function

There are D such inputs x_i that are used as input to this fourier basis function, so $f_i x$ is $\mathbb{R}^D \rightarrow \mathbb{R}$

Controlling Complexity of the Manifolds

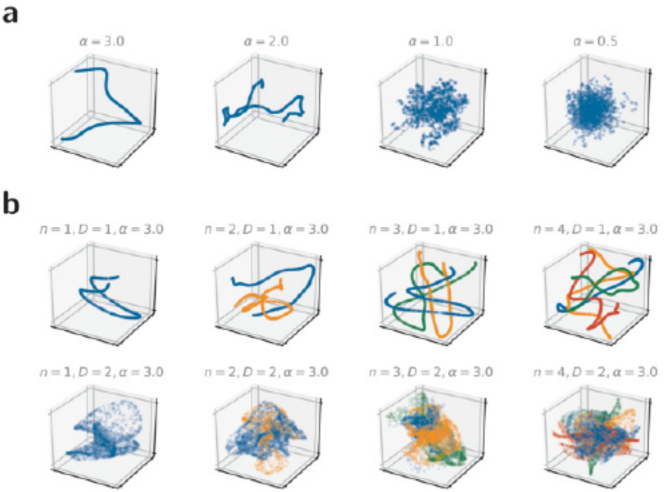


Figure: Manifolds

Spike Rasters

- The Co-ordinates of the points are obtained after sampling points from a $D - \mathit{dimensional}$ Hypercube
- The M co-ordinates correspond to the firing times of M individual input neurons

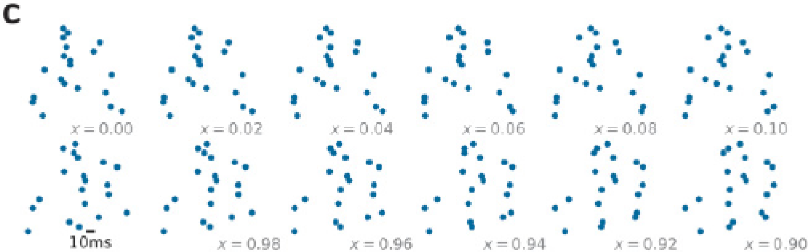


Figure: Spike Rasters for a 1D Manifold

Model Architecture

The Neuron Model is a simple Leaky Integrate and Fire model, with a decay constant β_{mem}

$$U_i^l[n+1] = \left[U_i^l[n](\beta_{mem}) + (1-\beta_{mem})I_i^l[n]\right](1 - S_i^l[n]) \tag{2}$$

On rearranging the terms one can see the LIF model induced

$$\delta U = -(1-\beta_{mem})(U_i^l[n]) + (1-\beta_{mem})I_i^l[n] \tag{3}$$

This looks like the LIF model that we studied earlier.

Decays and Synapses

Each constant β has the form

$$\beta \approx \exp(\frac{-\Delta t}{\tau}) \tag{4}$$

Synapses and current

$$\boxed{I_l[n+1] = \beta_{syn} I_l[n] + \sum_j W_j S_{l-1}[n] + \sum_k V_k S_l[n]} \tag{5}$$

Spikes and Threshold

1. The Spikes are passed as 0 - 1 (binary inputs)
2. If a Neuron Spikes, then at the next time-step, the Potential U is set to 0
3. The reset is efficiently achieved by using a Heaviside Step Function

$$\boxed{S_i^l[n] = \Theta(U_i^l[n] - 1)} \tag{6}$$

Note:

Note that Θ is Non-differentiable. This is where we choose to use the surrogate gradients instead of the true gradients

The Computational Graph

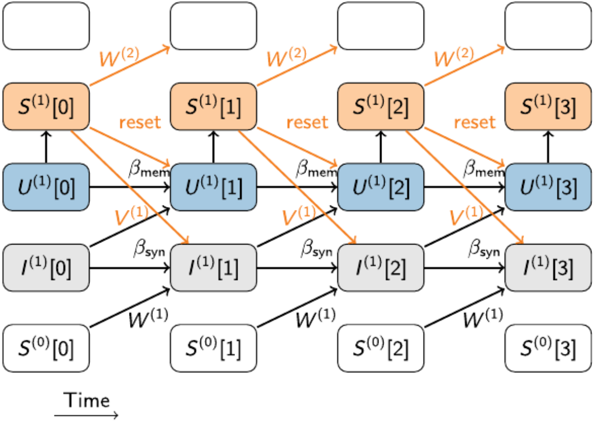


Figure: Computational Graph

Back Propagation Through Time

Since the loss function being used is a maximum of the readout activation over time (or an average), the gradients with respect to the loss will flow back to every timestep. We need to calculate $\frac{\partial L}{\partial W}$ and $\frac{\partial L}{\partial V}$'s

Readout Layers

The Readout layers do not spike, only the activation is accumulated over time. Which is then used as an argument to the Loss function

$$U_i^{out} = \beta_{out} U_i^{out} + (1 - \beta_{out}) I_i^{out}[n]$$

Surrogate Gradients

If we try to run the backpropagation, then we will get an inflow of gradients into every lower layer for the smaller time steps. We will have to calculate the gradient of the spiking function(heaviside step)

Surrogate Gradients

If we try to run the backpropagation, then we will get an inflow of gradients into every lower layer for the smaller time steps. We will have to calculate the gradient of the spiking function(heaviside step)

$$\boxed{S'(U_i[n]) \approx \sigma'(U_i[n] - 1)} \tag{7}$$

Surrogate Gradients

If we try to run the backpropagation, then we will get an inflow of gradients into every lower layer for the smaller time steps. We will have to calculate the gradient of the spiking function(heaviside step)

$$\boxed{S'(U_i[n]) \approx \sigma'(U_i[n] - 1)} \tag{7}$$

- SuperSpike $\frac{1}{1+\beta|x|^2}$
- Esser $max(0, 1 - \beta|x|)$
- Sigmoid Derivative $s(x)(1 - s(x))$

BPTT

Calculating ∇L involves calculating the gradients of $I_{out}[n]$ with respect to W. If we differentiate Equation [5], we will get a recurrence of the following form :-

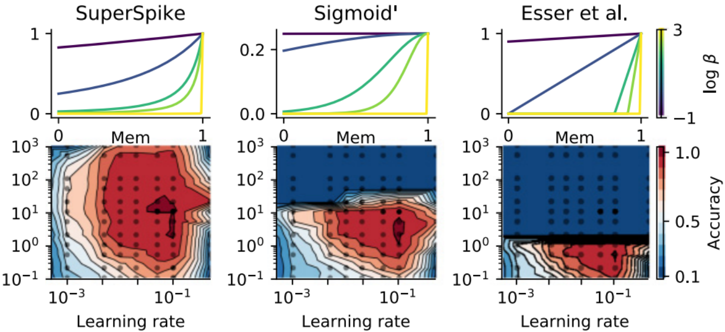
$$\nabla I_{n+1} = \nabla I_n + \sum_j W_j \frac{\partial S}{\partial W} \tag{8}$$

Scale of the SG

This shows the importance of the scale of the surrogate gradient, it may cause a vanishing/exploding gradient problem.

Robustness of SGL

- Robust to shape of surrogate derivative
  - Using different surrogate derivatives does not affect maximum performance.
  - Slope of surrogate derivatives does not affect maximum performance.
  - Changing derivative function or its slope changes the parameter space in which model gives good performance



Robustness of SGL

- Sensitive to scale of surrogate derivative
  - Surrogate gradient is usually normalized to 1.
  - Scale may determine whether gradients vanish or explode.
  - Empirical effect of scale is seen when recurrence is present and spike reset is part of BPTT - Large scale results in detrimental performance.
- Robust to loss function, different datasets and different data paradigms
  - No significant difference between "max" and "sum" readouts.
  - Consistent performance on MNIST images (converted to spike trails for inputs), and raw audio data (raw current input).

Robustness to Input Paradigm and Loss

- **Datasets**: Synthetic random manifolds

Robustness to Input Paradigm and Loss

- **Datasets**: Synthetic random manifolds
- MNIST (pixel values $\rightarrow$ spike latencies)

Robustness to Input Paradigm and Loss

- **Datasets**: Synthetic random manifolds
- MNIST (pixel values $\rightarrow$ spike latencies)
  - Each neuron fires either $0/1$ spikes for each input

Robustness to Input Paradigm and Loss

- **Datasets**: Synthetic random manifolds
- MNIST (pixel values $\rightarrow$ spike latencies)
  - Each neuron fires either 0/1 spikes for each input

Robustness to Input Paradigm and Loss

- **Datasets**: Synthetic random manifolds
- MNIST (pixel values $\rightarrow$ spike latencies)
  - Each neuron fires either 0/1 spikes for each input

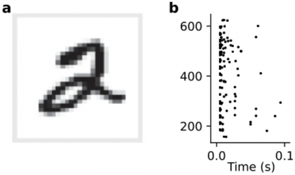


Figure: Spike rastor plot of an MNIST image

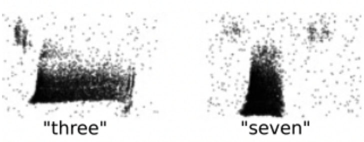


Figure: Spike rastor plots for spoken digits

Robustness to Input Paradigm and Loss

- **Loss**: Maximum over time, Summation over time
- No substantial difference in accuracy across datastes

Robustness to Input Paradigm and Loss

- **Loss**: Maximum over time, Summation over time
- No substantial difference in accuracy across datastes

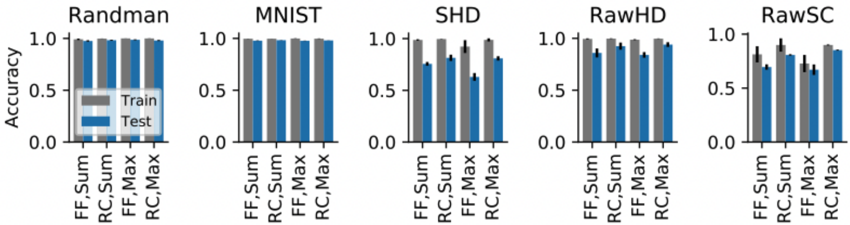


Figure: Accuracy across datasets. Sum vs Max. FF (Feed Forward), RC(Explicitly Recurrent)

Robustness to Input Paradigm and Loss

- **Loss**: Maximum over time, Summation over time
- No substantial difference in accuracy across datastes

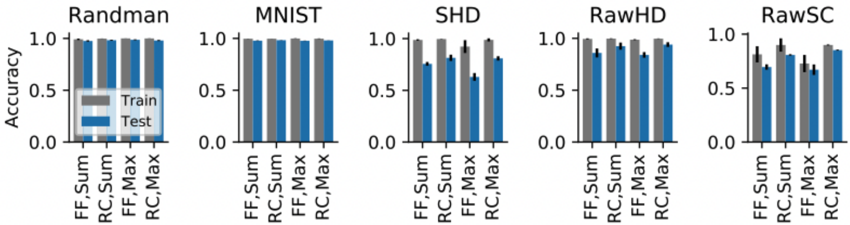


Figure: Accuracy across datasets. Sum vs Max. FF (Feed Forward), RC(Explicitly Recurrent)

- Reason for absence of affect: Input spike trains need to be longer for the effect of recurrent connections to emerge.

Performance on Current-based Inputs

- Inputs considered: spike-based

Performance on Current-based Inputs

- Inputs considered: spike-based
- Analog $\rightarrow$ spiking encoder could affect performance

Performance on Current-based Inputs

- Inputs considered: spike-based
- Analog $\rightarrow$ spiking encoder could affect performance
- Directly feed in current-based input

Performance on Current-based Inputs

- Inputs considered: spike-based
- Analog $\rightarrow$ spiking encoder could affect performance
- Directly feed in current-based input

Performance on Current-based Inputs

- Inputs considered: spike-based
- Analog $\rightarrow$ spiking encoder could affect performance
- Directly feed in current-based input

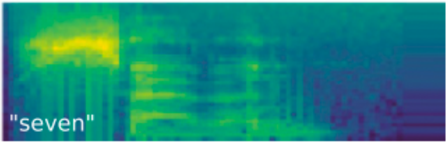


Figure: Raw audio to Mel-space spectrogram

- Reduced overfitting. Better accuracy than a FF network

Optimal Sparse Spiking Activity

- Neurons should not spike very often as that wastes energy.

$$g^{\mu}_{\text{upper}} = -\lambda_{\text{upper}} \left(\left[\frac{1}{M} \sum_{i}^{M} \zeta_i^{(l),\mu} - \nu_{\text{upper}} \right]_+ \right)^L$$

- Neurons should not be dormant.

$$g^{\mu}_{\text{lower}} = \frac{\lambda_{\text{lower}}}{M} \sum_{i}^{M} \left(\left[\nu_{\text{lower}} - \zeta_i^{(l),\mu} \right]_+ \right)^2$$

- There is a lower critical limit on the number of spikes, below which there is a drop in performance.

Optimal Sparse Spiking Activity

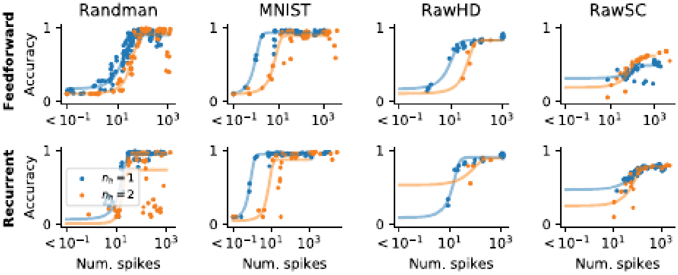


Figure: Performance relative to number of hidden layer spikes.

Optimal Sparse Spiking Activity

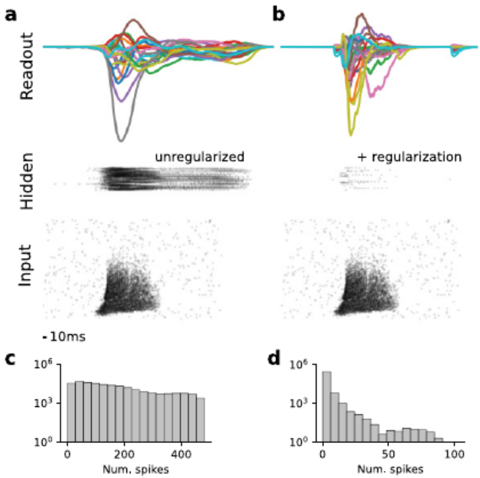


Figure: Comparison between regularized and unregularized spiking activity.

Conclusion

- Surrogate gradients should be appropriately normalized.
- Activity regularization leads to sparsity with 1-2 orders of magnitude fewer spikes, with no significant effect in performance.
- Future work
  - Optimal initialization of hidden layer weights.
  - Do the findings apply to Convolutional SNNs?
  - Why do surrogate gradients work well when ignoring spike reset in BPTT?