# Including STDP to eligibility propagation in multi-layer recurrent spiking neural networks.

Werner van der Veen
(w.k.van.der.veen.2@student.rug.nl)

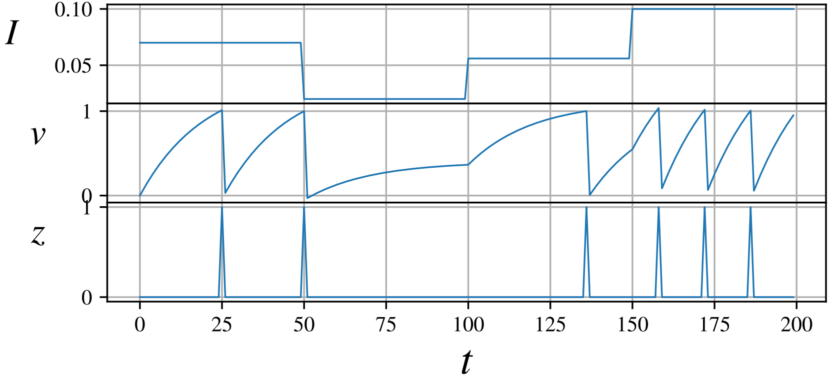
April 28, 2021

# Introduction – context

- Biological inspiration to AI
- ANNs and backpropagation
- Deep learning
  $\rightarrow$ high energy consumption
- Fundamental differences between DL and biological learning
  - Continuous vs. binary communication
  - Backpropagation/BPTT vs. learning signals

# Introduction – SNNs

- Binary spikes $\rightarrow$ no backpropagation
- Good performance, but not competitive with DL
- Leaky integrate-and-fire (LIF) neuron



# Introduction – neuromorphic computing

- Physical embedding of neural network in an analog medium
- Colocalized memory and computation
  $\rightarrow$ no backpropagation
- SNNs as ideal biologically inspired NC paradigm for efficient computation
- Analog SNNs are extremely fast and energy efficient
- Problem: learning rule must be *local* and *online*
  $\rightarrow$ revalue biological learning

# Introduction – biological learning

- Hebbian learning
  $\rightarrow$ runaway excitation
- STDP
  $\rightarrow$ how to teach?
- Learning signals: three-factor Hebbian learning, R-STDP
- Biological learning signals: neurotransmitters
  $\rightarrow$ credit assignment
- Eligibility traces

# Introduction – eligibility propagation

- Mathematical approximation to BPTT
- Local learning signals and eligibility traces
- Applicable to any SNN topology and multiple neuron models
- Also applicable to neuromorphic VLSIs
- Competitive with LSTMs on phone classification
- E-prop: only ALIF so far
- New research: STDP-LIF and Izhikevich neuron display STDP

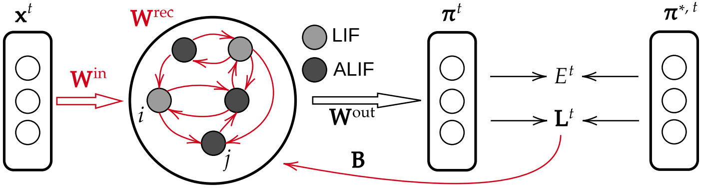
# Introduction – this research

Research question:

*Does including STDP-like behavior in e-prop lead to faster and more accurate phone classification?*

- Reproduce results of original e-prop paper
- Extend STDP-LIF to STDP-ALIF
- Experimentally verify the STDP-ALIF and Izhikevich neurons on phone classification task
- Examine effects of e-prop in multi-layer recurrent SNN

# Methods – technical framework (e-prop)

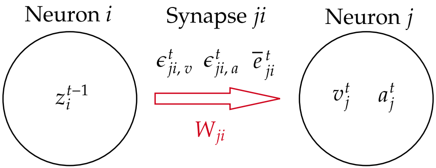



E-prop model $\mathcal{M} = \langle M, f \rangle$

$$\mathbf{h}_j^t = M\left(\mathbf{h}_j^{t-1}, \mathbf{z}^{t-1}, \mathbf{x}^t, \mathbf{W}_j\right) \tag{1}$$

$$z_j^t = f\left(\mathbf{h}_j^t\right) \tag{2}$$

# Methods – the ALIF neuron model

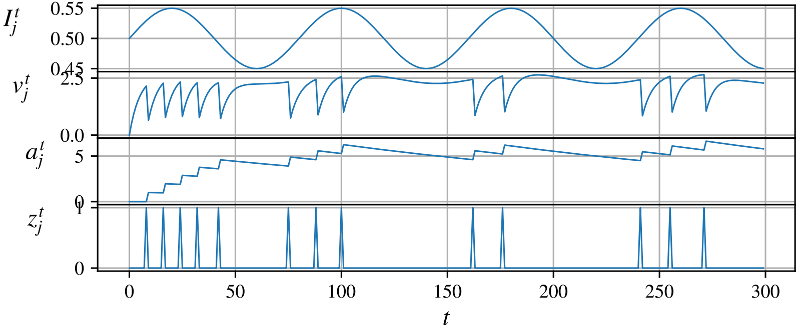


# Methods – the ALIF neuron model

$$z_j^t = H\left(v_j^t - v_{\text{th}} - \beta a_j^t\right) \tag{3}$$

$$v_j^{t+1} = \alpha v_j^t + \sum_{i \neq j} W_{ji}^{\text{rec}} z_i^t + \sum_i W_{ji}^{\text{in}} x_i^{t+1} - z_j^t v_{\text{th}} \tag{4}$$

$$a_j^{t+1} = \rho a_j^t + z_j^t \tag{5}$$



# Methods – e-prop

$$\frac{dE}{dW_{ji}} = \sum_t L_j^t \; e_{ji}^t$$

$$\frac{dE}{dW_{ji}} = \sum_t \frac{dE}{dz_j^t} \underbrace{\frac{\partial z_j^t}{\partial \mathbf{h}_j^t} \underbrace{\sum_{t \geq t'} \frac{\partial \mathbf{h}_j^t}{\partial \mathbf{h}_j^{t-1}} \cdots \frac{\partial \mathbf{h}_j^{t+1}}{\partial \mathbf{h}_j^{t'}} \cdot \frac{\partial \mathbf{h}_j^{t'}}{\partial W_{ji}}}_{\epsilon_{ji}^t}}_{e_{ji}^t}$$

# Methods – e-prop for ALIF

Hidden state:

$$\mathbf{h}_j^t = \begin{pmatrix} v_j^t \\ a_j^t \end{pmatrix} \tag{6}$$

Pseudo-derivative:

$$\psi_j^t = \gamma \max\left(0, 1 - \left|\frac{v_j^t - v_{\text{th}} - \beta a_j^t}{v_{\text{th}}}\right|\right) \tag{7}$$

Eligibility vector:

$$\begin{pmatrix} \epsilon_{ji,v}^{t+1} \\ \epsilon_{ji,a}^{t+1} \end{pmatrix} = \begin{pmatrix} \alpha \cdot \epsilon_{ji,v}^t + z_i^{t-1} \\ \psi_j^t \epsilon_{ji,v}^t + \left(\rho - \psi_j^t \beta\right) \epsilon_{ji,a}^t \end{pmatrix} \tag{8}$$

Eligibility trace:

$$e_{ji}^t = \psi_j^t \left(\epsilon_{ji,v}^t - \beta \epsilon_{ji,a}^t\right) \tag{9}$$

# Methods – e-prop weight update

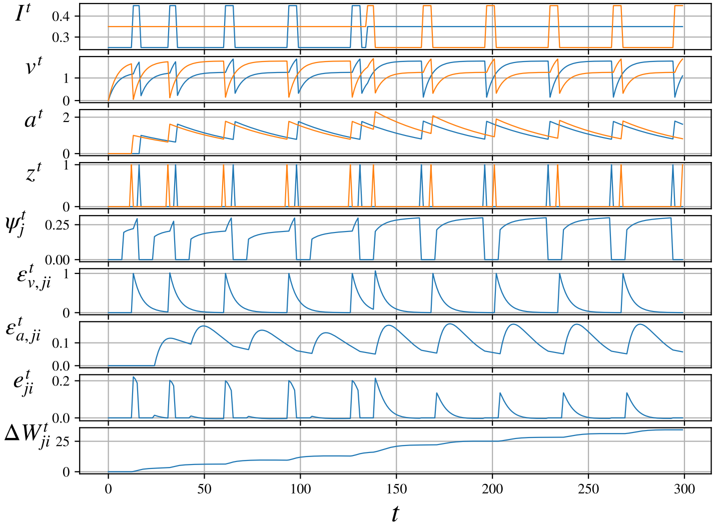
$$\Delta W_{ji} = -\eta \sum_t \underbrace{\sum_k B_{jk} \left(\pi_k^t - \pi_k^{*,t}\right)}_{=L_j^t} \underbrace{\sum_{t' \leq t} \kappa^{t'-t} e_{ji}^{t'}}_{\stackrel{\text{def}}{=} \bar{e}_{ji}^t} \tag{10}$$

where $\hat{y}_k^t = \kappa \hat{y}_k^{t-1} + \sum_j W_{kj}^{\text{out}} z_j^t$,
$y_k^t = \hat{y}_k^t + b_k$
and $\pi_k^t = \sigma_k\left(y_1^t, \ldots, y_K^t\right)$.

$$\Delta W_{kj}^{\text{out}} = -\eta \sum_t \left(\pi_k^t - \pi_k^{*,t}\right) \sum_{t' \leq t} \kappa^{t'-t} z_j^t \tag{11}$$

$$\Delta b_k = -\eta \sum_t \left(\pi_k^t - \pi_k^{*,t}\right). \tag{12}$$

# Methods – Visualizing the ALIF neuron



# Methods – STDP-ALIF

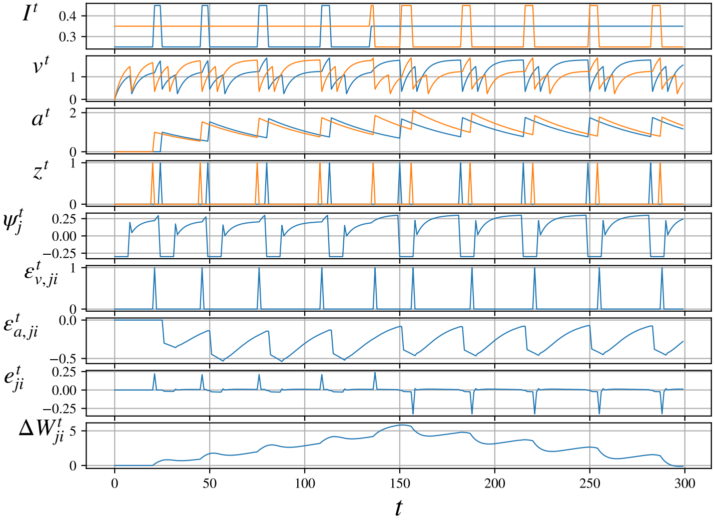
- New neuron model:

$$v_j^{t+1} = \alpha v_j^t + \sum_{i \neq j} W_{ji}^{\text{rec}} z_i^t + \sum_i W_{ji}^{\text{in}} I - z_j^t \alpha v_j^t - z_j^{t-\delta t_{\text{ref}}} \alpha v_j^t \tag{13}$$

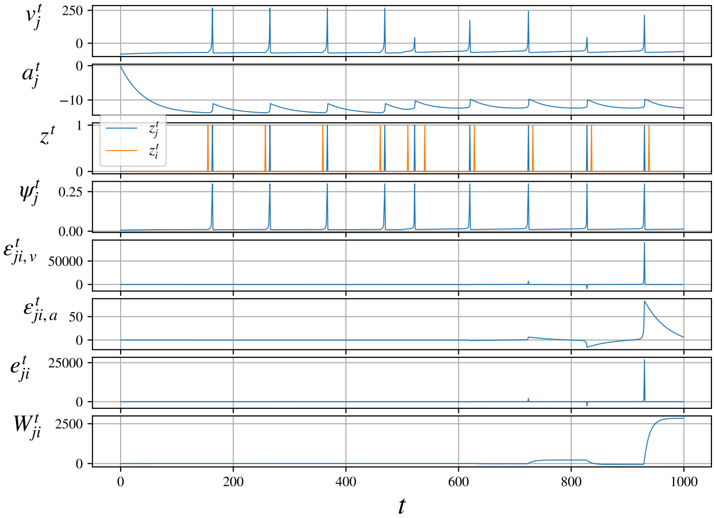
$$\psi_j^t = \begin{cases} -\gamma & \text{if } t - t_{z_j} < \delta t_{\text{ref}} \\ \gamma \max\left(0, 1 - \left|\frac{v_j^t - v_{\text{th}}}{v_{\text{th}}}\right|\right) & \text{otherwise} \end{cases} \tag{14}$$

- Decreases $v_j^t$ after refractory period, and clamps $\psi_j^t$ to $-\gamma$ instead of to 0.

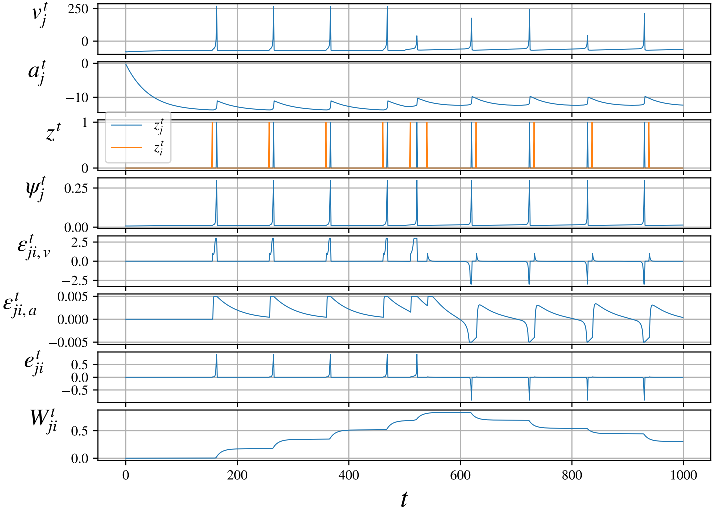
# Methods – STDP-ALIF



# Methods – Izhikevich



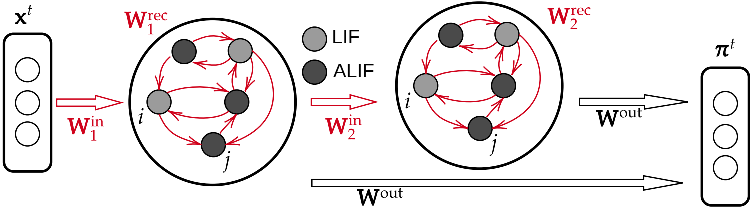
# Methods – Izhikevich fixed



# Methods – Multilayer

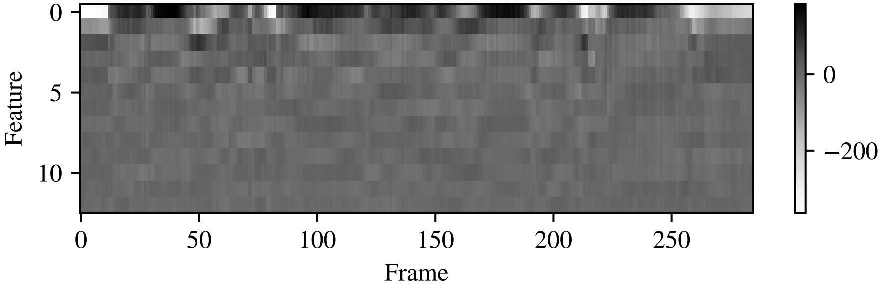
$$\mathbf{h}_{rj}^{t} = \begin{cases} M\left(\mathbf{h}_{rj}^{t-1}, \mathbf{z}_{r}^{t-1}, \mathbf{x}^{t}, \mathbf{W}_{rj}\right) & \text{if } r = 1 \\ M\left(\mathbf{h}_{rj}^{t-1}, \mathbf{z}_{r}^{t-1}, \mathbf{z}_{r-1}^{t}, \mathbf{W}_{rj}\right) & \text{otherwise,} \end{cases} \tag{15}$$

where $r \in [1 \mathinner{..} R]$.



# Methods – Task

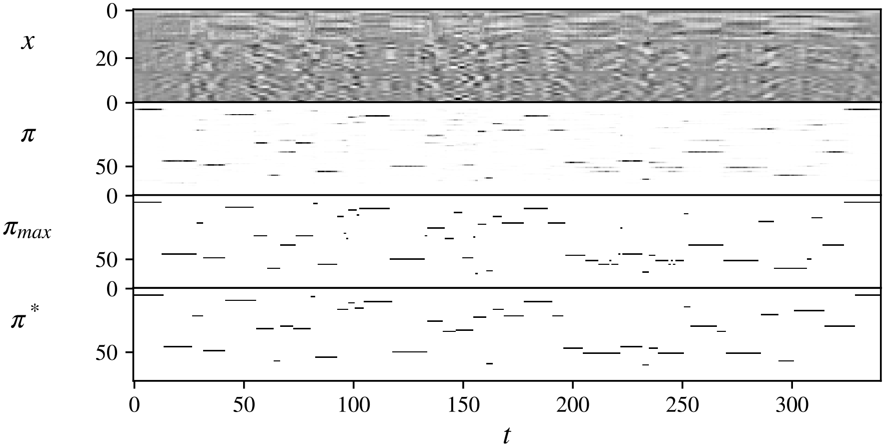
- Classify frame-wise phones from speech signals.
- There are 61 different class labels, and approximately 4000 speech sentences containing around 200-300 frames each.
- Speech is preprocessed into MFCCs (2–13) with their 1$^{st}$ and 2$^{nd}$ deltas, standardized per channel



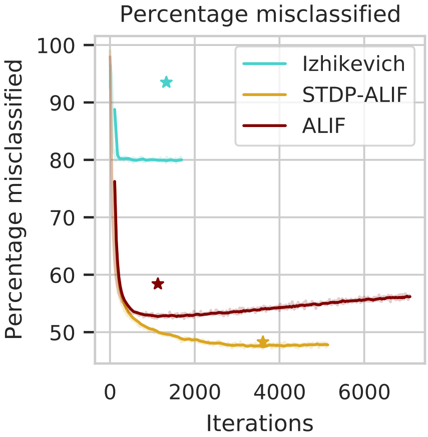
# Methods – Other settings

- Adam optimizer for e-prop
- Firing rate regularization
- L2 regularization

# Results – Example

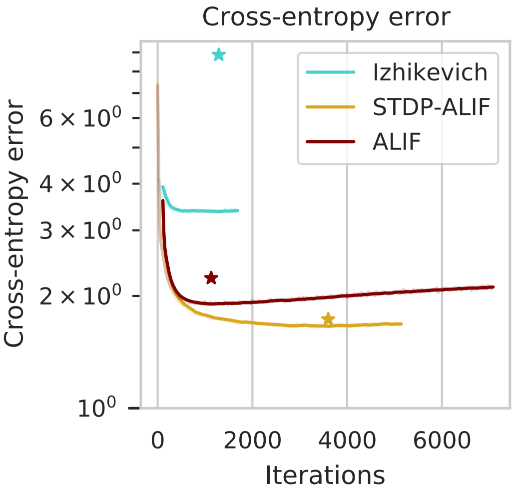


# Results – Accuracy per neuron type

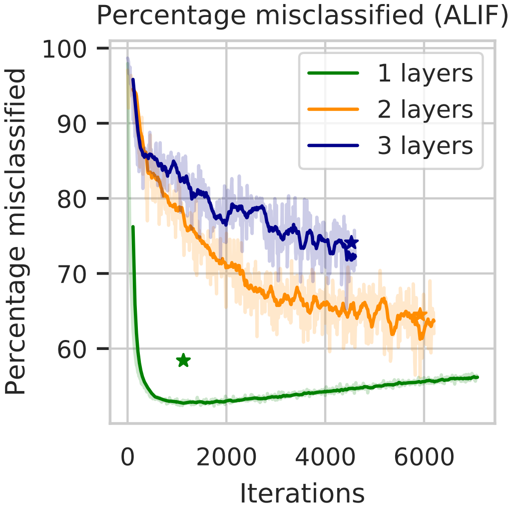



Test scores: ALIF = 58.4%, STDP-ALIF = 48.3%, Izhikevich = 93.5%

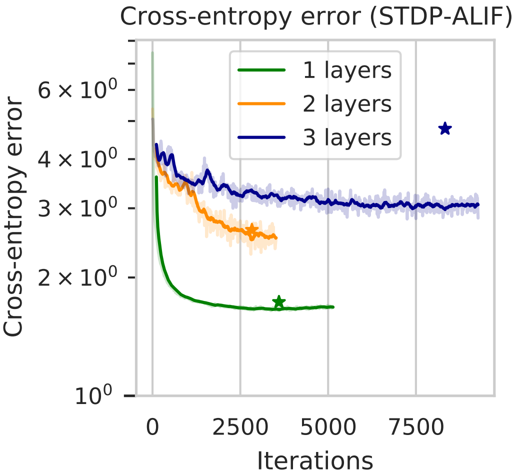
# Results – Cross-entropy per neuron type



# Results – Multi-layer effects



# Results – Multi-layer effects



# Discussion

- More layers = less efficient
- Including STDP-like behavior:
  - improves the classification performance of the e-prop model (but not always!)
  - more closely resembles biological learning.

# Discussion – future research

- Smoothen output
- Distributed parameters
- Custom connectivity graphs
- Dynamic pruning and growing
- Synaptic delay

# Conclusion

- Including STDP in e-prop improves classification performance and efficiency. $\rightarrow$ one of the best biologically plausible learning algorithms for SNNs.
- Good trade-off between running cost and performance
- More bioplausibility $\rightarrow$ easier in NC but not always better performance
- More research is needed on applying e-prop in other types of learning tasks, and many ideas exist that may improve performance even more.

*Thank you!*

$$\frac{dE}{dW_{ji}} = \sum_{t' \leq T} \frac{dE}{d\mathbf{h}_j^{t'}} \cdot \frac{\partial \mathbf{h}_j^{t'}}{\partial W_{ji}} = \sum_t \frac{dE}{dz_j^t} \cdot \left[\frac{dz_j^t}{dW_{ji}}\right]_{\text{local}} \tag{16}$$

$$\frac{dE}{d\mathbf{h}_j^{t'}} = \underbrace{\frac{dE}{dz_j^{t'}}}_{L_j^{t'}} \frac{\partial z_j^{t'}}{\partial \mathbf{h}_j^{t'}} + \frac{dE}{d\mathbf{h}_j^{t'+1}} \frac{\partial \mathbf{h}_j^{t'+1}}{\partial \mathbf{h}_j^{t'}} \tag{17}$$

$$\frac{dE}{dW_{ji}} = \sum_{t'} \left( L_j^{t'} \frac{\partial z_j^{t'}}{\partial \mathbf{h}_j^{t'}} + \left( L_j^{t'+1} \frac{\partial z_j^{t'+1}}{\partial \mathbf{h}_j^{t'+1}} + (\cdots) \frac{\partial \mathbf{h}_j^{t'+2}}{\partial \mathbf{h}_j^{t'+1}} \right) \frac{\partial \mathbf{h}_j^{t'+1}}{\partial \mathbf{h}_j^{t'}} \right) \cdot \frac{\partial \mathbf{h}_j^{t'}}{\partial W_{ji}} \tag{18}$$

$$\frac{dE}{dW_{ji}} = \sum_{t'} \sum_{t \geq t'} L_j^t \frac{\partial z_j^t}{\partial \mathbf{h}_j^t} \frac{\partial \mathbf{h}_j^t}{\partial \mathbf{h}_j^{t-1}} \cdots \frac{\partial \mathbf{h}_j^{t+1}}{\partial \mathbf{h}_j^{t'}} \cdot \frac{\partial \mathbf{h}_j^{t'}}{\partial W_{ji}} \tag{19}$$

$$\frac{dE}{dW_{ji}} = \sum_t \frac{dE}{dz_j^t} \underbrace{\frac{\partial z_j^t}{\partial \mathbf{h}_j^t} \underbrace{\sum_{t \geq t'} \frac{\partial \mathbf{h}_j^t}{\partial \mathbf{h}_j^{t-1}} \cdots \frac{\partial \mathbf{h}_j^{t+1}}{\partial \mathbf{h}_j^{t'}} \cdot \frac{\partial \mathbf{h}_j^{t'}}{\partial W_{ji}}}_{\boldsymbol{\epsilon}_{ji}^t}}_{e_{ji}^t} \tag{20}$$

$$v_j^{t+1} = \alpha v_j^t + \sum_{i \neq j} W_{ji}^{\mathsf{rec}} z_i^t + \sum_i W_{ji}^{\mathsf{in}} x_i^{t+1} - z_j^t v_{\mathsf{th}} \tag{21}$$

$$a_j^{t+1} = \rho a_j^t + z_j^t \tag{22}$$

$$z_j^t = H\left(v_j^t - v_{\mathsf{th}} - \beta a_j^t\right) \tag{23}$$

$$\frac{\partial v_j^{t+1}}{\partial v_j^t} = \alpha \tag{24}$$

$$\frac{\partial v_j^{t+1}}{\partial a_j^t} = 0 \tag{25}$$

$$\frac{\partial a_j^{t+1}}{\partial v_j^t} = \psi_j^t \tag{26}$$

$$\frac{\partial a_j^{t+1}}{\partial a_j^t} = \rho - \psi_j^t \beta \tag{27}$$

$$\frac{\partial z_j^t}{\partial \mathbf{h}_j^t} = \begin{pmatrix} \frac{\partial z_j^t}{\partial v_j^t} \\ \frac{\partial z_j^t}{\partial a_j^t} \end{pmatrix} = \begin{pmatrix} \psi_j^t \\ -\beta \psi_j^t \end{pmatrix} \tag{28}$$