

State Space Reduction for Spiking Neural Network Verification

via Filtration-Based Model Abstraction

Research Note — January 2026

1. Introduction

Probabilistic model checking provides formal guarantees for systems exhibiting stochastic behavior. When applied to Spiking Neural Networks (SNNs), we encode the network dynamics as a Discrete-Time Markov Chain (DTMC) and verify properties expressed in Probabilistic Computation Tree Logic (PCTL). However, the *state explosion problem* renders verification intractable for networks with more than a handful of neurons.

This note presents an approach based on *filtration* from modal logic to construct the smallest PCTL-equivalent model, dramatically reducing the state space while preserving all properties of interest.

2. Problem Statement

2.1. SNN as a DTMC

Consider an SNN with n neurons. Each neuron i has:

- Membrane potential $p_i \in [P_{\min}, P_{\max}] \subset \mathbb{Z}$
- Spike output $y_i \in \{0, 1\}$
- Refractory state $s_i \in \{0, 1, 2\}$ (optional)

The state space of the full model is:

$$|S| = \prod_{i=1}^n (|P_{\max} - P_{\min} + 1|) \cdot 2 \cdot 3$$

For a 6-neuron network with potential range $[0, 200]$ and refractory periods enabled:

$$|S| = (201 \cdot 2 \cdot 3)^6 \approx 1.8 \times 10^{\{18\}}$$

Even with reachability analysis reducing this to $\sim 10^6$ states, verification time remains prohibitive for iterative tasks such as weight learning.

2.2. Current Optimizations

We have implemented several engineering optimizations in the CogSpike system:

1. **Per-neuron potential bounds:** Computing P_{\max}^i based on incoming weights
2. **Disabled refractory periods:** Removing state variables when not needed
3. **Reduced threshold levels:** Using 4 instead of 10 probabilistic firing bands

These reduce the theoretical state space but do not exploit the fundamental *semantic equivalence* between states.

3. Theoretical Foundation

3.1. Bisimulation Equivalence

Two states s, s' in a Kripke structure \mathcal{M} are *bisimilar* (written $s \sim s'$) if:

1. They satisfy the same atomic propositions: $L(s) = L(s')$
2. For every transition $s \rightarrow t$, there exists $s' \rightarrow t'$ with $t \sim t'$ (and vice versa)
3. The transition probabilities to equivalence classes are preserved

The *bisimulation quotient* \mathcal{M}/\sim collapses all bisimilar states, yielding the smallest model that satisfies the same PCTL* formulas [1].

3.2. Filtration

Filtration, introduced in modal logic [2], constructs a finite model from a potentially infinite one by considering only distinctions relevant to a finite set of formulas Σ .

Definition. Given a Kripke model $\mathcal{M} = (W, R, V)$ and a subformula-closed set Σ , the *filtration* of \mathcal{M} through Σ is:

$$\mathcal{M}^f = (W^f, R^f, V^f)$$

where:

- $W^f = W / \approx_\Sigma$ with $w \approx_\Sigma w' \iff \forall \varphi \in \Sigma : \mathcal{M}, w \models \varphi \iff \mathcal{M}, w' \models \varphi$
- $V^f([w]) = V(w) \cap \text{Props}(\Sigma)$
- R^f satisfies: if wRw' then $[w]R^f[w']$ (smallest filtration)

Theorem (Filtration Theorem). For all $\varphi \in \Sigma$ and $w \in W$:

$$\mathcal{M}, w \models \varphi \iff \mathcal{M}^f, [w] \models \varphi$$

The size of \mathcal{M}^f is bounded by $2^{|\Sigma|}$, independent of $|W|$. [2]

4. Application to SNN Verification

4.1. Membrane Potential Equivalence

For PCTL properties concerning spike events (e.g., $P_{=?}[F^{\leq T} \text{spike}_i]$), the membrane potential matters only insofar as it determines *firing probability*.

With k threshold levels, we define:

$$p \approx_k p' \iff \text{fire_prob}(p) = \text{fire_prob}(p')$$

This partitions the potential range into $k + 1$ equivalence classes:

$$\begin{aligned} C_0 &= \{p : p < \theta_1\} && \text{(no firing)} \\ C_j &= \{p : \theta_j \leq p < \theta_{j+1}\} && \text{(fires with prob } j/k) \\ C_k &= \{p : p \geq \theta_k\} && \text{(certain firing)} \end{aligned}$$

4.2. State Space Reduction

Proposition. For a network with n neurons and k threshold levels, the filtrated state space has size:

$$|S^f| = (k + 1)^n \cdot 2^n$$

compared to the original:

$$|S| = |P|^n \cdot 2^n \cdot 3^n$$

where $|P|$ is the potential range size.

Example. For the 6-neuron double-diamond network with $|P| = 201$ and $k = 4$:

- Original: $201^6 \cdot 2^6 \cdot 3^6 \approx 4 \times 10^{18}$ (theoretical)
- With disabled refractory: $201^6 \cdot 2^6 \approx 4 \times 10^{15}$
- With filtration ($k = 4$): $5^6 \cdot 2^6 \approx 10^6$

This is a **reduction factor of 10^9** over the disabled-refractory model.

4.3. Transition Probability Preservation

The critical requirement is that the quotient model preserves transition probabilities. Define $P^f : S^f \times S^f \rightarrow [0, 1]$ as:

$$P^f([s], [t]) = \sum_{t' \in [t]} P(s, t')$$

For our SNN abstraction, if state s has potential class C_j^i for neuron i :

- Neuron fires with probability j/k
- Upon firing, potential resets to C_0
- Upon not firing, potential evolves based on inputs

The key observation is that the *next potential class* depends only on:

1. Current potential class (not exact value)
2. Input spike pattern
3. Weight configuration

This ensures the quotient is well-defined.

5. Proposed Implementation

5.1. Abstract PRISM Model Generation

Instead of generating:

```
p2 : [0..200] init 0;
[tick] p2 >= threshold4 -> 1.0 : (y2'=1) & (p2'=0);
```

Generate the abstracted model:

```
pClass2 : [0..4] init 0; // 5 equivalence classes
[tick] pClass2 = 4 -> 1.0 : (y2'=1) & (pClass2'=0);
[tick] pClass2 = 3 -> 0.75 : (y2'=1) & (pClass2'=0)
      + 0.25 : (y2'=0) & (pClass2'=nextClass2);
```

5.2. Computing Class Transitions

The function $\text{nextClass} : C \times 2^I \rightarrow \mathcal{D}(C)$ maps current class and input pattern to a distribution over next classes:

$$\text{nextClass}(C_j, \bar{x}) = \begin{cases} C_{\text{clamp}(j+\delta)} & \text{if neuron doesn't fire} \\ C_0 & \text{if neuron fires} \end{cases}$$

where δ is determined by the weighted input sum and leak rate.

6. Research Contributions

This approach offers several potential contributions:

1. **Formal framework:** Applying filtration theory to probabilistic SNN models
2. **Correctness proof:** Showing PCTL property preservation for spike-related queries
3. **Implementation:** Automated quotient model generation from SNN specifications
4. **Evaluation:** Empirical comparison of state space size and verification time

7. Conclusion

The state explosion problem in SNN verification can be addressed not just through engineering optimizations, but through principled model abstraction based on modal logic filtration. By identifying states equivalent with respect to firing behavior, we can construct quotient models with dramatically reduced state spaces while provably preserving PCTL properties.

Future work includes implementing the quotient generator, proving correctness formally, and evaluating on benchmark networks.

Bibliography

- [1] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.
- [2] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, vol. 53. in Cambridge Tracts in Theoretical Computer Science, vol. 53. Cambridge University Press, 2001.