

Spiking Neural Networks, Spiking Neural P Systems, and Their Application

Claudio Zandron

Università Milano-Bicocca, Italy

Outline

- 1 Introduction
- 2 Spiking Neural Networks
- 3 Spiking Neural P Systems
- 4 SNN Applications
- 5 SNPS Applications
- 6 Comparison and Future Directions

- **Spiking Neural Networks (SNNs)** and **Spiking Neural P Systems (SNPSs)** are advanced computational models
- Inspired by biological processes of the human brain
- Use discrete spikes to transmit information between neurons
- Offer more energy-efficient and biologically plausible approach compared to traditional neural networks
- Particularly well-suited for resource-constrained environments

- In biological systems, neurons communicate by emitting electrical signals known as **spikes**
- Spikes are transmitted through synapses
- This mechanism allows neurons to influence the activity of neighbor neurons
- A spike generated by one neuron can be transmitted as input to another
- Traditional ANNs often demand high computational power and are difficult to scale

Why SNNs and SNPSs?

Energy Efficiency

- Significantly reduced energy consumption
- Sparse, asynchronous operation
- Neurons can spike independently without all components being active

Biological Plausibility

- More realistic modeling of neural communication
- Temporal dynamics of neural activity
- Synaptic plasticity for learning and adaptation

Spiking Neural Networks - Overview

- Unlike conventional ANNs that rely on continuous activation values
- SNNs transmit discrete spikes over time
- Capture temporal dynamics of neural activity
- Neuron's action potential determined by a threshold
- When threshold exceeded, triggers emission of a spike
- More efficient information encoding based on timing and frequency of spikes

SNN Formal Definition

A Spiking Neural Network is formally defined as:

- Finite set V of spiking neurons
- Set $E \subseteq V \times V$ of synapses
- Weight $W_{u,v} \geq 0$ for each synapse $(u, v) \in E$
- Response function $s_{u,v} : \mathbb{R}^+ \rightarrow \mathbb{R}$
- Threshold function $\theta_v : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ for each neuron $v \in V$

SNN Components

Input Neurons

- Subset $V_{input} \subset V$
- Firing times provided externally
- Not determined by the model

Output Neurons

- Subset $V_{output} \subset V$
- Firing times constitute network output

Hidden Neurons

- All other neurons $v \in V \setminus V_{input}$
- Firing times determined by model rules

- Biological neurons don't always fire deterministically
- Stochastic version models this uncertainty
- Difference $P_v(t) - \theta_v(t - t')$ determines firing probability
- Neuron may not fire even when $P_v(t) - \theta_v(t - t') > 0$
- May fire spontaneously when $P_v(t) - \theta_v(t - t') < 0$
- Doesn't significantly impact computational properties

Spiking Neural P Systems - Overview

- Combine principles of SNNs with membrane computing
- Set of neurons arranged as nodes in directed graph
- Each neuron can contain any number of spikes (denoted by symbol a)
- Can fire under specific conditions based on internal spike count
- Spikes travel along graph arcs (synapses) to connected neurons
- System evolves from initial configuration according to defined rules

SNPS Formal Definition

A Spiking Neural P system of degree $m \geq 1$:

$$\Pi = (O, \sigma_1, \dots, \sigma_m, syn, i_0)$$

Where:

- ① $O = \{a\}$ is the singleton alphabet (spike)
- ② $\sigma_1, \dots, \sigma_m$ are neurons: $\sigma_i = (n_i, R_i)$
- ③ $syn \subseteq \{1, 2, \dots, m\} \times \{1, 2, \dots, m\}$ (synapses)
- ④ $i_0 \in \{1, 2, \dots, m\}$ indicates output neuron

Spiking Rules

$E/a^c \rightarrow a^k; d$ where:

- E is regular expression over $\{a\}$
- $c, k \geq 1$ and $d \geq 0$
- Consumes c spikes, produces k spikes with delay d

Forgetting Rules

$a^s \rightarrow \lambda$ where:

- $s \geq 1$
- Can only be applied if neuron contains exactly s spikes
- No spiking rules are enabled
- Removes all s spikes

SNPS Operation

- Operates synchronously with global discrete clock
- Spiking rule enabled when neuron contains n spikes such that $a^n \in L(E)$ and $n \geq c$
- When fired: consumes c spikes, prepares k spikes for delivery with delay d
- During delay: neuron enters refractory period (closed to incoming spikes)

Rule Application Principles

- 1 **Sequentiality at neuron level:** Only one rule per neuron
- 2 **Maximal parallelism at system level:** All enabled neurons fire simultaneously

SNPS Configuration and Computation

Configuration

$C_t = \langle k_1/t_1, \dots, k_m/t_m \rangle$ where:

- k_i : number of spikes in neuron σ_i
- $t_i \geq 0$: remaining refractory period ($t_i = 0$ means open)

Computation

- Sequence of transitions between configurations
- Starts from initial state
- Halts when all neurons are open and no rules enabled
- Deterministic if regular expressions of rules have disjoint languages

SNN Applications Overview

Despite training challenges, SNNs have found applications in:

- **Image Recognition:** MNIST dataset, character recognition
- **Pattern Classification:** Various datasets and problems
- **Medical Applications:** Diagnostics and analysis
- **Robot Control:** Autonomous systems
- **Neuromorphic Hardware:** Specialized computing platforms

Early SNN Applications

MNIST Dataset Recognition

- Spike-driven synaptic plasticity with integrate-and-fire neurons
- 2000 input neurons for pattern classification
- 300 classes of preprocessed LaTeX characters
- NIST character dataset subset

Unsupervised Learning

- Linear encoder/decoder with sparse non-linearity
- Generates "stroke detectors" for handwritten digits
- Produces Gabor-like filters for natural images
- Achieved lower error rate than best previous MNIST results

Temporal Coding

- Each neuron fires exactly one spike per stimulus
- Firing order encodes information
- Test accuracy: 97.4% on MNIST, 99.2% on Caltech Face/Motorbike

Center-Surround Structure

- Inspired by retinal receptive fields
- Integrate-and-Fire neuron model
- Iris dataset: 96.33% accuracy with 60 input neurons
- MNIST: 90.5% accuracy with 600 input neurons

COVID-19 Detection with SNNs - Motivation

Research Challenge

- Automated diagnosis of COVID-19 from chest X-rays
- Need for energy-efficient solutions for resource-constrained devices
- Real-time processing requirements in healthcare settings

Why SNNs for Medical Imaging?

- Suitable for IoT and edge computing applications
- Potential implementation on compact neuromorphic hardware (Intel Loihi)
- Biologically plausible computational model
- Significant reduction in energy consumption vs traditional ANNs

Dataset and Preprocessing

Extensive and Augmented COVID-19 Dataset

- Total: 8,734 chest X-rays from multiple public sources
- 3,728 COVID-19 positive cases
- 5,006 non-COVID cases
- Publicly available under CC BY 4.0 license

Preprocessing Steps

- Images resized to 256×256 pixels
- Converted to grayscale
- Pixel standardization (divided by 255)
- Split: 70% training, 14% validation, 16% test

Proposed Convolutional SNN Architecture

Network Design Principles

- Emphasis on simplicity for easy implementation
- Minimalistic design maximizing performance
- Reduced computational overhead
- Optimized for neuromorphic hardware deployment

Architecture Components

- Three convolutional layers (3×3 filters: 8, 64, 128)
- One max-pooling layer (4×4 , stride 2)
- Three fully connected layers (128, 64, 8 neurons)
- Final layer: 2 neurons + softmax for classification
- Activation: Spiking ReLU function

Implementation Framework

- Implemented using Nengo framework
- Spike frequency treated as continuous variable for optimization
- Hybrid approach: forward computation as frequencies
- Gradient-based optimization despite discrete spike nature

Training Parameters

- Optimizer: Adam with categorical cross-entropy loss
- Training: 20 epochs, batch size 32
- Validation set used to prevent overfitting
- Final evaluation on unseen test set

Classification Results

Performance Metrics

- **Accuracy:** 95%
- **Sensitivity:** 93.6% (correct COVID-19 identification)
- **Specificity:** 96% (correct non-COVID classification)

Key Findings

- Competitive performance with traditional ANNs
- Results comparable to complex deep learning architectures
- High discriminative ability between COVID-19 and non-COVID cases

Energy Consumption Analysis

Energy per Inference (Joules)

- **CPU (Intel i7-4960X):** 1.43×10^{-1} J/inference
- **GPU (Nvidia GTX Titan Black):** 5.00×10^{-1} J/inference
- **SNN on CPU:** 3.53×10^{-2} J/inference
- **Neuromorphic Hardware (Intel Loihi):** Significant further reduction

Energy Efficiency Gains

- $\sim 14\times$ less energy than DNNs on GPU
- $\sim 4\times$ less energy than DNNs on CPU
- Maintains competitive accuracy (95%)
- Ideal for battery-powered and edge devices

Comparison with State-of-the-Art

Comparison with ANNs

Network	AUC
VGG-16	0.95
CB-STM-RENet	0.99
DarkNet	0.97
ResNet	0.96
Proposed SNN	0.988

Key Advantages

- Comparable accuracy to best ANNs
- Significantly lower energy consumption
- Simpler architecture enabling compact hardware deployment
- Suitable for resource-constrained environments

Recent SNN Developments

Neuromorphic Hardware

- Advanced platforms foster SNN research
- Complex classification tasks
- Robot control applications
- Medical applications

Performance Characteristics

- Results close to traditional ANNs
- **Significantly lower energy consumption**
- Non-linear neural spiking networks explored
- Effective for temporal information processing

SNPSs have diverse applications across multiple domains:

- **Classification Tasks:** Multi-class, high-dimensional problems
- **Medical Imaging:** Brain tumor detection, organ segmentation
- **Power Systems:** Microgrid management, energy distribution
- **Language Processing:** Dialect recognition, letter recognition
- **Graph Theory:** Node classification in large networks
- **Cybersecurity:** Malware detection, threat classification
- **Federated Learning:** Decentralized machine learning

SNPS Classification Applications

Layered SNPS with Supervised Learning

- Multi-layer network structure
- Evaluated on UCI machine learning repository
- MNIST dataset experiments
- Demonstrated effectiveness for classification problems

Advantages

- Energy-efficient properties
- Sparse and asynchronous activity
- Neurons active only when accumulating enough spikes
- Many neurons remain inactive, reducing energy consumption

Brain Tumor Detection

- Magnetic Resonance Imaging (MRI) analysis
- Threshold segmentation approach based on SNPSs
- Results comparable to (sometimes better than) optimization algorithms

Other Medical Tasks

- Organ segmentation using deep dynamic spiking neural P systems
- Cancer gene selection with adaptive optimization
- Survival time prediction for glioblastoma patients
- Various medical diagnostics applications

Distributed Energy Systems

- Fuzzy spiking neural systems for coordinated control
- Optimizes energy distribution across interconnected microgrids
- Improves efficiency and stability

Enhanced Framework

- Integration of autapses in SNPS model
- More adaptive control of multi-microgrid systems
- Better energy management
- Real-time decision-making in dynamic power networks

Chinese Dialect Tone Recognition

- Gated SNPSs applied to tone recognition
- High accuracy in recognizing tonal variations
- Outperformed traditional machine learning methods

English Letter Recognition

- Adaptive optimization SNPSs
- Effective classification of handwritten letters
- Complex pattern recognition in natural language processing domain

Node Classification

- Hierarchical SNPSs with weighted connections
- Graph-based node classification problems
- Exploits temporal dynamics of spiking neurons

Performance

- Efficiently processes large-scale graph data
- High accuracy in node classification
- Applications in network analysis and social media analytics
- Handles inherently interconnected data

Cyber-SNPSs

- Detection of malicious activities
- Malware and phishing attack classification
- Superior performance in detecting cyber threats

Advantages

- Fewer training epochs required
- Higher accuracy than traditional ML approaches
- Fast and accurate threat detection
- Enhanced security for digital infrastructures

SNPS Federated Learning

Layered SNPSs in Federated Learning

- Decentralized machine learning paradigm
- Faster convergence than other federated learning algorithms
- Higher accuracy achieved
- Privacy-preserving machine learning

Benefits

- Data processed locally on decentralized devices
- Protects sensitive information
- Efficient learning across distributed networks
- Attractive solution for privacy-preserving ML

Modular Spiking Neural P Systems - Motivation

Challenge: Multi-Class Image Classification

- Classification problems with hundreds or thousands of classes
- High structural complexity of networks
- Need for efficient approaches to handle variable quality images

Solution: Modular Approach

- Divide general problem into smaller sub-problems
- Each module specializes in learning one specific class
- Multiple instances of same learning module in system
- Energy-efficient variant of SNNs suitable for small neuron counts

Dataset Characteristics

- 8,189 images of flowers from 102 UK species
- RGB images with variable quality
- 40 to 258 images per class
- Large variations: scale, pose, lighting conditions

Dataset Challenges

- Classes with large intra-class variations
- Similar classes difficult to distinguish
- Subjects not always centered
- Different image shapes and resolutions
- Split: 1020 training, 6149 test, 1020 validation

System Structure

- 102 independent learning modules (one per flower class)
- Each module connected to entire dataset
- Learns only positive examples of assigned class
- Other images treated as negative examples

Three-Layer Module Design

- 1 **Input layer:** Receives and transmits encoded spikes
- 2 **Distinguish layer:** Separates spikes encoding 0 from 1
- 3 **Encoding layer:** Stores learned information via created synapses

Encoding and Learning Process

Spike Encoding

- RGB values (0-255) mapped to binary using threshold δ
- Values $< \delta$ mapped to a (one spike)
- Values $\geq \delta$ mapped to a^2 (two spikes)
- Images converted to vectors for parallel processing

Learning Mechanism

- **Positive examples:** Create synapses to encoding layer
- **Negative examples:** Remove existing synapses
- Neurons in encoding layer activate circularly
- One reading of training set completes learning
- No synapse strengthening/weakening required

MSNP Classification Results

Performance Metrics

- Best weighted accuracy: 85% (class 1, RGB images)
- Accuracy range across all classes: 51.7% to 94.9%
- Mean accuracy: 68.5% (RGB), 60.9% (grayscale)
- 150×150 resolution with 0.6 downscaling factor optimal

Performance Distribution (RGB)

- 30 classes: 50-60% accuracy
- 28 classes: 60-70% accuracy
- 25 classes: 70-80% accuracy
- 16 classes: 80-90% accuracy
- 3 classes: $>90\%$ accuracy

Impact of Image Quality on Performance

Factors Affecting Accuracy

- **Poor performance causes:**
 - Non-centered subjects
 - Excessive background visible
 - Subject not shown in entirety
 - Significant pose variations
 - Mostly cropped subjects

High Performance Conditions

- Centered subjects with similar poses
- Good quality images with clear subjects
- Minimal background interference
- Consistent presentation across class images

SNNs vs SNPSs Comparison

Similarities

- Both are third-generation neural networks
- Rely on spike-based communication
- Energy-efficient properties
- Adaptability and computational capability
- Simulate biological processes with high fidelity

Key Differences

- **SNPSs**: Grounded in formal language-theoretic framework
- **SNNs**: Primarily inspired by biological systems
- **SNNs**: Support various encoding schemes for continuous inputs/outputs
- **SNPSs**: Encode information through spike numbers or discrete delays

Conclusions

Key Achievements

- SNNs and SNPSs represent significant advancements in neural computation
- More biologically plausible than traditional ANNs
- Demonstrated effectiveness across diverse application domains
- Energy-efficient and computationally capable

Impact and Promise

- Enhance artificial intelligence capabilities
- Provide insights into human brain processes
- Promising tools for resource-constrained environments
- Open exciting opportunities for biologically inspired computing
- Potential for revolutionary advances in neuromorphic computing

Challenges and Future Research

Current Challenges

- Development of training algorithms for SNNs and SNPSs
- Discrete nature of spikes complicates learning
- Integration into broader computational frameworks

Future Research Directions

- Improving training methods addressing discrete spike nature
- Advancing neuromorphic hardware to exploit full potential
- Exploring hybrid models combining SNNs and SNPSs strengths
- Applications in cryptography and fault detection
- New insights into human brain computational processes