CS 5463: Survey-based Term Project
(Appendix)

Topic: A Survey of Spiking Neural Networks and Their Learning Strategies

Classification of the Results:

1. **Foundations of Spiking Neural Networks (SNNs)**
   Focus: Biological inspiration for neural computation, the rise of SNNs as the third generation of neural networks, and some key terminologies about SNNs.

   Primary Papers:
   - (Maass, 1997)

   Secondary Papers:
   - (Ghosh-Dastidar & Adeli, 2009)

2. **Neuron Models and Signal Encoding in SNNs**
   Focus: Popular neuron models and common input encoding schemes for SNNs.

   Primary Papers:
   - (Niu et al., 2023)
   - (Tan et al., 2020)

   Secondary Papers:
   - (J. Wu et al., 2019)
   - (Rathi et al., 2023)
   - (Cruz-Albrecht et al., 2012)
   - (Zhang et al., 2019)
   - (Hodgkin & Huxley, 1952)
   - (Yamazaki et al., 2022)
   - (Diehl & Cook, 2015)
   - (Rueckauer et al., 2017)
   - (Iaboni & Abichandani, 2024)
   - (Thorpe & Gautrais, 1998)
   - (Buzsáki, 2006)

3. **Learning and Training Methods in SNNs**
   Focus: Covers how SNNs learn using supervised, unsupervised, and indirect learning approaches.

   Primary Papers:
   - (Diehl & Cook, 2015)
   - (Lee et al., 2016)

   Secondary Papers:
   - (Neftci et al., 2019)

- (Sengupta et al., 2019)
- (H. Wu et al., 2021)
- (Y. Wu et al., 2018)
- (Jin et al., 2018)
- (Rueckauer et al., 2017)
- (Rueckauer & Liu, 2018)
- (Rathi et al., 2020)
- (Iaboni & Abichandani, 2024)
- (Legenstein et al., 2005)
- (Markram et al., 2012)
- (Song et al., 2000)
- (Pietrzak et al., 2023)
- (Ruf & Schmitt, 1997)
- (Bohte et al., 2000).
- (Lee et al., 2016)
- (McKennoch et al., 2006)
- (Wang et al., 2020)
- (Xu et al., 2013)
- (Masquelier & Thorpe, 2007)
- (Hunsberger & Eliasmith, 2015)
- (Kingma & Ba, 2014)
- (Zenke & Ganguli, 2018)
- (Simonyan & Zisserman, 2014)
- (Szegedy et al., 2016)
- (Lecun et al., 1998)
- (Rathi et al., 2023)

4. **Challenges and Future Research in SNNs**

Focus: Highlights current limitations in SNNs and explores future research opportunities in the field.

Primary Papers:
- (Kudithipudi et al., 2025)

Secondary Papers:
- (Rathi et al., 2023)
- (Pfeiffer & Pfeil, 2018)
- (Tavanaei et al., 2019)
- (Bagheri et al., 2018)
- (Liang et al., 2020)
- (Ioffe & Szegedy, 2015)
- (Sanaullah et al., 2023)
- (Lee et al., 2016)
- (H. Wu et al., 2021)
- (Y. Wu et al., 2018)
- (Jin et al., 2018)
- (Neftci et al., 2019)

- (Diehl & Cook, 2015)
- (Legenstein et al., 2005)
- (Masquelier & Thorpe, 2007)
- (Sengupta et al., 2019)
- (Rueckauer et al., 2017)
- (Rueckauer & Liu, 2018)
- (Rathi et al., 2020)

Detailed Annotated Bibliography:

1. **(Maass, 1997)**

   Summary: This paper compares the computing capacity of spiking neuron networks (third-generation models) to older neural networks built with McCulloch-Pitts and sigmoidal neurons. It shows that spiking neurons are more efficient, requiring fewer neurons for certain tasks, and better reflect biological neural processing by using spike timing for information encoding. The paper highlights their ability to perform fast, analog computations and provides lower bounds on neuron counts, which shows their superiority over traditional models for specific functions.

   Methodology: The author formally defines spiking neuron models (types A and B) and uses complexity theory to compare their efficiency with first- and second-generation neural networks. He proves theorems that establish neuron count lower bounds and shows example functions that spiking neurons compute more efficiently.

   Strengths: The paper provides theoretical guarantees showing that spiking neurons possess superior computational capacity compared to traditional neurons. It also offers biologically relevant examples and rigorous mathematical proofs.

   Weaknesses: The analysis is largely theoretical and may not reflect practical training or implementation challenges. The results assume idealized neurons and may not hold under noisy or real-world conditions.

   Challenges and Future Directions: The paper highlights the need for further studies into noise-resilient and real-time computation models with SNNs. Future work could explore training algorithms and hardware implementations.

   Applications: These findings are useful for designing neuromorphic hardware and models that mimic fast biological processing.

2. **(Ghosh-Dastidar & Adeli, 2009)**

   Summary: This paper analyzes the advancement of SNNs, emphasizing their significance as a next step in the evolution of neural network models. It discusses how SNNs work like biological neurons by encoding information with precise spike timing, which allows for more dynamic and biologically realistic models. The paper also explores various learning algorithms, both supervised and unsupervised, that have been developed for SNNs, which shows how useful they could be for solving difficult temporal pattern recognition problems.

   Methodology: The authors categorize neuron models from first to third generation and compare learning methods like STDP, SpikeProp, and Multi-SpikeProp. They explain how spiking neurons encode information through temporal integration of postsynaptic potentials and describe network architectures for both unsupervised clustering and supervised classification.

Strengths: The paper offers a historical and conceptual overview of the field, linking biological plausibility with computational models. It explains neuron dynamics in detail and discusses applications from theory to EEG classification.

Weaknesses: It focuses heavily on older benchmarks and small-scale problems, limiting relevance for today's large-scale SNN applications. Most models discussed are not yet hardware-optimized or scalable.

Challenges and Future Directions: Major challenges include the computational cost and scalability of biologically realistic models. The authors suggest developing efficient encodings and adaptive synapse modeling as future directions.

Applications: SNNs are applicable in EEG classification, time-series prediction, and biologically inspired pattern recognition.

3. **(Yamazaki et al., 2022)**

   Summary: The paper reviews SNNs, which are based on biological neurons and use spikes for communication, making them energy-efficient and suitable for real-time applications. It discusses various neuron models like Hodgkin-Huxley and Leaky Integrate-and-Fire, along with learning methods such as Spike-Timing-Dependent Plasticity (STDP) and ANN-to-SNN conversion. The paper highlights SNN applications in computer vision, including object detection and optical flow estimation, and in robotics for tasks like navigation and locomotion. Although SNNs have potential, they encounter obstacles in training and performance on large-scale datasets in comparison to traditional deep learning models.

   Methodology: The authors categorize existing work by neuron and synapse models, learning algorithms (e.g., STDP, surrogate gradients, ANN-to-SNN), and application domains. They also provide case studies across vision and robotic tasks, detailing training and inference methods.

   Strengths: The review offers broad coverage across biological foundations, models, and applications. It connects theory with practical neuromorphic implementations on datasets like ImageNet and environments like Loihi.

   Weaknesses: There is less empirical benchmarking of the methods across standard datasets. Some discussions, like spike encoding and STDP variants, remain more descriptive than comparative.

   Challenges and Future Directions: The authors call for better training methods, architecture search for SNNs, and improving SNN performance on large datasets like ImageNet.

   Applications: SNNs are applicable in energy-efficient real-time systems for vision and robotics, especially on neuromorphic hardware.

4. **(Sanaullah et al., 2023)**

   Summary: The paper explores various mathematical models of SNNs to replicate the behavior of neurons, focusing on their performance, computational efficiency, and biological plausibility. It compares models like LIF, NLIF, AdEx, HH, and others using a synthetic dataset, measuring classification accuracy and performance loss. The AdEx model demonstrated the highest level of accuracy, while the HH model was the most biologically plausible but computationally expensive. The study emphasizes the significance of selecting the appropriate model for specific tasks and recommends future research directions, such as hardware implementation and robustness testing.

   Methodology: The paper compares several SNN models by evaluating their performance, accuracy, and computational efficiency using a synthetic dataset. Each model's classification accuracy and performance loss were measured, providing an in-depth knowledge of their performance under similar situations.

   Strengths: The paper provides an in-depth comparison of various SNN models, highlighting their computational efficiency and biological plausibility. The use of a synthetic dataset ensures consistency in evaluating all models under the same conditions, which makes it easier to draw direct comparisons.

   Weaknesses: The dependency on a synthetic dataset limits the generalization of the results to more difficult real-world data. Additionally, the scalability of these models in large-scale tasks was not explored.

   Challenges and Future Directions: The research showed a significant obstacle in the selection of the most appropriate SNN model for classification tasks, which involves balancing accuracy and performance loss. Future work could explore hardware implementations and robustness testing for the models.

   Applications: The results can help guide the selection of SNN models for real-world classification tasks, such as pattern recognition.

5. **(Wang et al., 2020)**

   Summary: The paper reviews supervised learning algorithms for SNNs, focusing on their ability to handle temporal and spatiotemporal data with biologically plausible spiking neurons. It categorizes algorithms based on network architecture (single-layer, multilayer feed-forward, and recurrent SNNs) and learning mechanisms (gradient descent, spike train convolution, synaptic plasticity, etc.). The paper evaluates the performance of these algorithms using qualitative criteria such as spike train learning ability, online/offline processing, locality, stability, and applicability to different neuron models. It also suggests future research guidelines, such as hardware implementation, hybrid SNNs, and online education, and proposes a classification scheme for supervised learning algorithms.

Methodology: The authors organize and analyze existing supervised SNN learning algorithms based on architecture type, such as single-layer, feedforward, and recurrent models, and evaluate them using criteria like learning ability and structural adaptability. They also compare performance across spike train lengths and firing rates through simulations.

Strengths: This review provides a clear taxonomy and comparative evaluation of SNN learning methods. It also identifies biologically inspired approaches and practical learning rules that scale to different network types.

Weaknesses: The review focuses more on classification and less on regression or sequence learning. It includes limited benchmarking with standardized tasks beyond MNIST variants.

Challenges and Future Directions: The authors call for more research on online learning, hybrid models, hardware support, and stability in deeper SNNs with evolving structures.

Applications: The insights support the development of adaptive SNNs for pattern recognition in edge AI and neuromorphic systems.

6. **(Diehl & Cook, 2015)**

Summary: The paper presents a biologically plausible spiking neural network (SNN) for unsupervised digit recognition by spike-timing-dependent plasticity (STDP). The network consists of conductance-driven synapses, lateral restriction, and adjustable thresholds. Without the use of labeled data, it obtains an accuracy of 95% on the MNIST dataset. The authors demonstrate the robustness of the network by testing four different STDP rules, showing that the combination of mechanisms allows the network to learn prototypical inputs effectively. The network demonstrates consistent performance scaling as the number of neurons increases. It offers consistent results across different STDP learning rules, suggesting its potential for broader applications in neuroscience and machine learning.

Methodology: The authors used leaky integrate-and-fire neurons with conductance-based synapses. STDP was employed to learn synaptic weights between the input layer and excitatory neurons. They applied lateral inhibition by connecting each excitatory neuron to an inhibitory neuron, which in turn inhibited all other excitatory neurons. An adaptive threshold was used to balance firing rates. Inputs were encoded as Poisson-distributed spike trains derived from pixel intensities. They trained and tested their model on the MNIST dataset using the BRIAN simulator.

Strengths: The model is biologically realistic and does not use labels during training. It shows robustness across different learning rules. The approach shows consistent performance scaling as the number of neurons increases.

Weaknesses: Although the model is biologically plausible, it performs slightly lower than some rate-based models. Furthermore, performance depends on careful tuning of inhibitory connections and adaptive thresholds.

Challenges and Future Directions: A challenge is achieving high performance while keeping biological plausibility. The authors suggest expanding the architecture to deeper layers and exploring localized receptive fields for better generalization.

Applications: This model can be applied to low-power neuromorphic hardware for real-time image recognition. It is also useful in adaptive systems like spiking vision processors.

7. **(Pietrzak et al., 2023)**

   Summary: The paper reviews various learning approaches for SNNs, which focus on their computational complexities and performance on standard hardware like CPUs and GPUs. It divides learning algorithms into three groups, analyzing their efficiency and memory usage: backpropagation, spike-timing-dependent plasticity (STDP), and ANN-SNN conversion. The study finds that while SNNs are energy-efficient on neuromorphic hardware, they are less efficient on traditional hardware due to the need for multiple time-step simulations. The paper concludes that ANN-SNN conversion methods currently achieve the best performance on standard datasets such as CIFAR-10 and MNIST, but SNNs still face challenges in execution speed and hardware availability.

   Methodology: The authors classify and compare STDP, backpropagation, and ANN-SNN conversion methods by analyzing their memory use, execution speed, and accuracy on standard datasets such as CIFAR-10 and MNIST using CPUs and GPUs.

   Strengths: The paper provides a unified comparison of SNN training approaches based on realistic hardware limitations. It also presents empirical results with benchmarks, offering practical insights for researchers.

   Weaknesses: It primarily focuses on feedforward networks and simple datasets. The scope excludes recurrent architectures and advanced neuromorphic tasks.

   Challenges and Future Directions: Challenges include limited hardware availability and the inefficiency of SNNs on synchronous platforms. Future directions involve optimizing learning rules and hardware-software codesign.

   Applications: This study is relevant to selecting efficient SNN training methods for CPU/GPU deployment in embedded systems and robotics.

8. **(Tavanaei et al., 2019)**

   Summary: This paper discusses the latest advances in training deep SNNs, comparing supervised and unsupervised methods. It shows how difficult it is to train SNNs because of the non-differentiability of spike-based activation functions, which complicates the use of backpropagation. The paper discusses various architectures, such as spiking convolutional neural networks (CNNs) and recurrent SNNs. The paper further describes how they performed on tasks such as image and speech recognition. It indicates that SNNs are more power-efficient but still lag behind traditional deep neural networks in accuracy.

Methodology: The authors compare supervised and unsupervised methods for training deep SNNs, particularly addressing challenges like the non-differentiability of spike-based activation functions. They discuss spiking CNNs and recurrent SNNs and show their potential in image recognition tasks.

Strengths: The paper provides an in-depth overview of deep SNNs and offers insights into their architectures and performance in practical tasks. Its focus on power efficiency is particularly valuable for hardware implementations.

Weaknesses: Despite their power efficiency, SNNs still lag behind traditional DNNs in accuracy, especially for complicated tasks like image and speech recognition. It is still challenging to train deep SNNs because spike-based activation functions are not differentiable, which makes backpropagation harder to use.

Challenges and Future Directions: The paper suggests that future research should focus on improving SNN training methods and exploring hybrid models that combine SNNs and DNNs for better performance, especially in the field of neuroscience.

Applications: Recently SNNs have been used for many different things, like image processing, speech recognition, and medical diagnosis, which offers advantages in power-efficient systems and spatio-temporal data processing.

9. **(Neftci et al., 2019)**

Summary: This paper discusses how difficult it is to train SNNs, since they are binary and dynamic, particularly in deep architectures with hidden layers. To address these issues, the authors introduce Surrogate Gradient (SG) methods, which enable the nonsmooth spiking nonlinearity to be continuously relaxed. It enables gradient-based optimization. These methods are applied in various contexts, such as feedback alignment, local three-factor learning rules (e.g., SuperSpike), and spike-time-based learning, offering efficient and scalable training solutions for SNNs. The article highlights the potential of SG methods to bridge machine learning, neuroscience, and neuromorphic computing, while emphasizing the need for further research to optimize activation functions and generalize to continuous-time dynamics.

Methodology: The authors first reformulate SNNs as a special case of RNNs and then apply surrogate gradient methods to allow gradient-based learning. They use approximations of the spiking nonlinearity with smooth functions like fast sigmoid, exponential, or piecewise-linear to enable backpropagation.

Strengths: SG methods allow the training of deep SNNs with hidden layers, solving spatial and temporal credit assignment problems. These approaches can be implemented in modern ML frameworks and support both online and offline learning.

Weaknesses: SGs introduce bias due to approximation and may lead to vanishing gradients with some activation choices. They often lack theoretical grounding and require more empirical validation in complex networks.

Challenges and Future Directions: Further work is needed on optimal surrogate function design and extending SG methods to continuous-time dynamics in biologically plausible networks.

Applications: SG methods can power real-time, low-power learning in neuromorphic devices and edge AI systems.

10. (Lee et al., 2016)

Summary: The paper introduces a novel technique to train deep SNNs using backpropagation, addressing the challenge of non-differentiable spike events by processing membrane potentials as separate signals. The authors propose a framework that includes fully connected and convolutional SNNs, with techniques for weight initialization, error normalization, and regularization to stabilize training. The method is tested on the MNIST and N-MNIST datasets, and it obtains the highest level of accuracy for SNNs, with results comparable to conventional deep neural networks. The approach demonstrates the potential for SNNs to achieve high accuracy with fewer computational operations, which makes them appropriate for event-driven applications that use little power.

Methodology: The authors reformulate membrane potentials using low-pass filtered spike signals and treat spike-time discontinuities as noise for differentiability. They introduce spiking versions of fully connected and convolutional networks, supported by weight initialization, threshold regularization, and lateral inhibition via winner-take-all (WTA) circuits. Training is performed using SGD or ADAM directly on spikes, and the backpropagation equations are modified to suit LIF neurons. The models are tested on event-based datasets without conversion from ANNs.

Strengths: The paper successfully trains deep SNNs from scratch using true spike-based inputs, achieving 98.77% on MNIST and 98.66% on N-MNIST. It also reduces computational cost significantly compared to ANN equivalents. The method handles exploding/vanishing gradients well with proposed error normalization.

Weaknesses: Training requires layer-specific tuning of thresholds and parameters such as α and η. The approach does not generalize well to hardware yet and assumes availability of detailed spike timing statistics.

Challenges and Future Directions: Future work includes adapting the training framework to recurrent SNNs and scaling it to more complex tasks or larger event-driven datasets.

Applications: This method supports training event-driven SNNs suitable for real-time applications on neuromorphic hardware using dynamic visual sensors.

11. (**Y. Wu et al., 2018**)

Summary: The paper proposes an STBP (Spatio-Temporal Backpropagation) for training SNNs, which includes both temporal and spatial dynamics to improve performance. The authors present an iterative LIF model suited to gradient-based training. They tackle the non-differentiability of spike activity by estimating its derivative using several types of curves. The STBP algorithm is tested on MNIST and N-MNIST datasets. It obtains state-of-the-art accuracy without requiring complex training techniques. The results indicate that incorporating temporal dynamics significantly enhances SNN performance, making it more biologically plausible and hardware-friendly for neuromorphic computing.

Methodology: The authors developed a recursive form of the LIF model to support gradient-based optimization in both space and time domains. They introduced an STBP algorithm that performs error backpropagation by unfolding the spiking neuron states in space (layers) and time (timesteps). To address spike non-differentiability, they propose four approximations (rectangular, polynomial, sigmoid, and Gaussian curves) for surrogate gradients. Training is done using SGD and Adam without complex techniques like error or weight normalization.

Strengths: The method achieves 98.89% accuracy on MNIST and 98.78% on N-MNIST using simple architectures and no pretraining. It outperforms many state-of-the-art models while avoiding complex techniques. The use of spatio-temporal gradients improves robustness and generalization.

Weaknesses: The method still requires careful tuning of hyperparameters like decay rate and curve steepness. Although powerful, the algorithm has only been tested on relatively small datasets and may not yet scale well to larger tasks.

Challenges and Future Directions: Future work includes extending STBP to larger datasets like CIFAR-10 and more temporally complex tasks like voice recognition and accelerating training on hardware.

Applications: STBP is suitable for efficient SNN training in neuromorphic systems using event-based data from sensors like DVS.

12. (**Jin et al., 2018**)

Summary: The paper introduces an HM2-BP (hybrid macro/micro-level backpropagation) method for deep SNN training. S-PSPs (Spike-train-level post-synaptic potentials) are used to capture temporal dynamics at the micro-level, while rate-coded errors are backpropagated at the macro-level to guide learning. By directly calculating the rate-coded loss function’s gradient, this technique enables effective training of deep SNNs. The method obtains higher accuracy on datasets like MNIST, N-MNIST, EMNIST, and T146 Speech, outperforming existing SNN training methods.

Methodology: The authors compute S-PSPs to model neuron interactions at the micro level. They use a decoupled approximation to combine spike timing and rate in a differentiable way, enabling end-to-end training with macro-level error backpropagation.

Strengths: The technique gets state-of-the-art accuracy on multiple datasets by employing spike-only data. It avoids spike smoothing and ANN conversion while supporting deep SNN training.

Weaknesses: It depends on fixed thresholds and Poisson spike encodings, which reduce flexibility. Gradient approximation through decoupling may affect accuracy in complex tasks.

Challenges and Future Directions: The authors suggest expanding to more complex networks and making the method hardware-compatible for real-time neuromorphic applications.

Applications: This method is useful for spiking speech recognition and visual recognition on energy-efficient neuromorphic chips.

13. (H. Wu et al., 2021)

Summary: The paper introduces a novel backpropagation method called Accumulated Spiking Flow Backpropagation (ASF-BP) to train SNNs. The method reduces computational complexity by aggregating the inputs and outputs of spiking neurons across time steps instead of relying on spike trains. It also introduces an adaptive mechanism to adjust scale factors, reflecting real neuron dynamics. Experiments on MNIST, CIFAR10, and CIFAR10-DVS datasets demonstrate that ASF-BP obtains state-of-the-art performance while significantly reducing training time compared to other methods.

Methodology: ASF-BP replaces spike-train-based gradients with accumulated spiking flows and builds an equivalent network. It then applies a single-loop backpropagation based on these accumulated values, using adaptive scale factors to match actual neuron behavior.

Strengths: It achieves 99.65% on MNIST and 91.35% on CIFAR10, with up to 6× faster training. It avoids the vanishing gradient problem and simplifies backward computations.

Weaknesses: The method loses temporal precision, which can impact results on neuromorphic datasets. The mathematical model is less effective for LIF neurons in deeper networks.

Challenges and Future Directions: The authors plan to extend ASF-BP to improve temporal resolution and support deeper LIF-based SNNs more reliably.

Applications: ASF-BP is suitable for fast, energy-efficient training of SNNs in real-time pattern recognition and edge devices.

14. (Sengupta et al., 2019)

Summary: The paper introduces a conversion technique from deep Analog Neural Networks (ANNs) to SNNs using a novel weight-normalization approach called Spike-Norm. This technique ensures minimal accuracy loss during conversion by considering the actual spiking behavior of SNNs, unlike previous methods that relied solely on ANN activations. The authors show that their method works on challenging datasets like CIFAR-10 and ImageNet through deep architectures like VGG and ResNet, getting state-of-the-art SNN performance. Additionally, they indicate that SNNs exhibit increased sparsity in neural activity as network depth increases. This method can help event-driven neuromorphic devices save a lot of energy.

Methodology: The authors used a data-driven weight normalization strategy during ANN-to-SNN conversion, where neuron thresholds are adjusted according to the behavior of actual SNN spikes. They applied this Spike-Norm method layer by layer, starting from Poisson-encoded inputs and propagating spike trains through deep networks like VGG-16 and ResNet. The model operates without bias terms or batch normalization and uses dropout for regularization.

Strengths: The method achieves minimal accuracy loss, even on large datasets like ImageNet. It supports deeper networks and leads to greater sparsity in spiking activity, reducing compute and energy.

Weaknesses: The method requires per-layer tuning of thresholds and specific architecture constraints (such as no biases or batch normalization), which might limit flexibility.

Challenges and Future Directions: Future work could include extending Spike-Norm to architectures that include bias or exploring different spiking neuron models for even better conversion accuracy.

Applications: This method is ideal for large-scale image recognition tasks using low-power neuromorphic chips.

15. (Rueckauer et al., 2017)

Summary: This paper presents a conversion strategy from traditional analog deep networks to spiking neural networks to classify images. The conversion process allows nearly arbitrary CNN architectures to be transformed into SNNs. The authors show how to transform well-known CNN designs (such as VGG-16 and Inception-v3) into SNNs. They obtain the best results on the MNIST, CIFAR-10, and ImageNet datasets. SNNs' main advantage is their ability to balance classification error rate and quantity of operations, which allows for significant reductions in computational cost compared to traditional ANNs. For example, the SNN version of LeNet on MNIST achieves a 2x reduction in operations with only a slight increase in error rate.

Methodology: The authors developed a theoretical basis for converting ANN activations to equivalent SNN spike rates. They replaced ANN components with spike-based counterparts such as softmax, batch normalization, and max-pooling. They introduced a robust normalization scheme based on percentiles to improve firing rate estimates and reduce error. The conversion was applied to full-scale architectures like VGG-16 and Inception-v3 using their custom SNN toolbox.

Strengths: The method generalizes to a wide range of architectures and preserves classification performance. It achieves near-lossless conversion on CIFAR-10 and MNIST, with reductions in operations by up to 2x.

Weaknesses: Deep networks require longer simulation times due to accumulated approximation errors across layers. In larger models, energy savings are less significant compared to smaller networks.

Challenges and Future Directions: The authors aim to refine spike encoding to reduce transient dynamics and enhance performance on deeper layers. Future work includes adapting the approach for deployment on neuromorphic chips.

Applications: The method is suited for low-power, real-time vision systems using neuromorphic hardware.

16. (Rueckauer & Liu, 2018)

Summary: This article proposes a technique to convert Analog Neural Networks (ANNs) to Spiking Neural Networks (SNNs) utilizing a temporal coding scheme focused on the time-to-first-spike (TTFS). The TTFS method represents ANN neuron activations as inverse spike times in SNNs, reducing the number of spikes and computational costs significantly. The method achieves a 7-10X reduction in operations on the MNIST dataset with less than a 1% drop in accuracy in comparison to the original ANN. Three variants of the TTFS method are proposed: a baseline version, one with dynamic thresholds, and another using a clamped ReLU activation function, all of which improve efficiency while maintaining accuracy.

Methodology: The authors trained a LeNet-5 ANN on MNIST and converted it to an SNN using TTFS encoding. In the base model, neurons spike once, and the timing represents the inverse of their activation value. The dynamic threshold variant adjusts firing thresholds based on incoming spikes, while the clamped version retrains the ANN using a modified ReLU function to minimize the impact of long-latency spikes.

Strengths: The method reduces operations by up to 10 times with minimal accuracy loss. It avoids rate coding's high spike count and does not require many simulation steps.

Weaknesses: The baseline method can miss important delayed spikes. Dynamic thresholds and clamped ReLU improve accuracy but require retraining or more complex updates.

Challenges and Future Directions: Future work includes extending this method to larger datasets and improving compatibility with neuromorphic hardware platforms.

Applications: This method is well-suited for neuromorphic platforms targeting low-power image recognition and other energy-efficient embedded tasks.

17. (Rathi et al., 2020)

Summary: The paper presents a hybrid approach to training SNNs by integrating artificial-to-spiking network conversion with spike-timing-dependent backpropagation (STDB). The approach initializes SNN weights and thresholds using a converted ANN, which is then optimized via STDB with a novel surrogate gradient that depends on spike timing. This method minimizes the inference time steps by 10-25x compared to purely converted SNNs, achieving similar accuracy while significantly lowering training complexity. The approach shows its efficiency across CIFAR-10, CIFAR-100, and ImageNet datasets, obtaining 65.19% top-1 accuracy on ImageNet in only 250 time steps.

Methodology: The authors begin by training an ANN, which is later converted to an SNN using threshold balancing. Fine-tuning of the resulting SNN is then performed using STDB, a surrogate gradient method that uses the time since a neuron's last spike. The forward pass accumulates membrane potentials, and gradients are computed at each time step using spike timing. Dropout is used instead of batch normalization, and average pooling is preferred to preserve information across layers.

Strengths: The hybrid method achieves high accuracy with fewer time steps, reducing latency and energy. It also cuts down training epochs and allows using deeper architectures like VGG16 and ResNet34.

Weaknesses: The method requires constraints like no bias terms and fixed thresholds. It also needs a pretrained ANN, which adds to the initial training load.

Challenges and Future Directions: Further research can explore training SNNs from scratch using spike timing and test the method on more tasks beyond image classification.

Applications: The approach is ideal for large-scale neuromorphic systems aimed at energy-efficient image processing tasks.

18. (Kudithipudi et al., 2025)

Summary: This paper explores neuromorphic computing, a method that takes design inspiration from the brain to develop effective computing infrastructure, with a focus on those applications that have power limitations. It draws attention to the fundamental characteristics, such as distributed and hierarchical structures, sparsity, neuronal scalability, and asynchronous communication, which enable energy-efficient and real-time processing. The paper also explores the challenges in scaling these systems, including hardware/software co-design, integration with conventional systems, and the need for standardized benchmarks and tools. Applications in computer vision, robotics, and neuroscience are discussed, emphasizing the potential for neuromorphic systems to revolutionize AI and machine learning. The paper concludes by outlining open questions and future directions for the field, including the development of large-scale test beds and lifelong learning systems.

Methodology: The paper reviews the progression of neuromorphic systems, categorizing key architectural features that support scalability. It identifies system-level design elements and co-design principles needed to evolve from lab prototypes to production-ready neuromorphic platforms.

Strengths: It provides a detailed roadmap for scaling neuromorphic computing with both near- and long-term challenges. It also links technological innovations to practical applications across fields.

Weaknesses: The paper lacks comparative benchmarking data and formal performance metrics across platforms. It also does not offer implementation details for the proposed integration strategies.

Challenges and Future Directions: Challenges include hardware-software interoperability, standardized benchmarking, and scaling to brain-sized simulations. Future work should also address usability by non-experts through abstraction layers.

Applications: The findings are relevant to energy-efficient AI, edge computing, robotics, neuroscience, and scientific simulations.

19. (Bagheri et al., 2018)

Summary: The paper analyzes the adversarial sensitivity of SNNs and proposes a robust training mechanism to enhance their resilience. The study models a two-layer SNN architecture featuring both rate- and time-based encoding and rate- and first-to-spike decoding strategies. The authors demonstrate that SNNs are vulnerable to white-box adversarial attacks, where small perturbations in the input spike trains can significantly degrade classification accuracy. To mitigate this, they introduce a robust training method that incorporates adversarial examples during training, improving the SNN's performance under adversarial conditions. The results show that first-to-spike decoding is more resilient to certain types of attacks compared to rate decoding and that robust training significantly enhances the SNN's robustness to adversarial perturbations.

Methodology: The authors use a probabilistic GLM-based two-layer SNN trained via maximum likelihood. They generate adversarial examples using flip, add, and remove attacks and incorporate them in SGD-based training to improve model robustness.

Strengths: The method improves test accuracy under strong adversarial attacks by up to 42%. It also reveals key differences in vulnerability between decoding and encoding strategies.

Weaknesses: It is limited to two-layer networks and requires full knowledge of model parameters for adversarial sample generation.

Challenges and Future Directions: Future work should extend the robust training method to deeper SNNs and more general attack scenarios.

Applications: This method is ideal for improving the robustness of low-power SNNs in adversarial environments such as IoT edge devices.

20. (Liang et al., 2020)

Summary: This paper investigates adversarial attacks on SNNs, highlighting difficulties like gradient-related limitations such as gradient incompatibility and vanishing gradients. The authors introduce two components, such as a Gradient-to-Spike (G2S) converter and a Restricted Spike Flipper (RSF), to generate spike-compatible adversarial examples. Experiments show 99%+ attack success rates on benchmarks like MNIST and CIFAR-10, demonstrating SNNs' robustness compared to ANNs. The work highlights the importance of loss function and firing threshold tuning for effective attacks.

Methodology: The authors use backpropagation-through-time to compute spatio-temporal gradients in SNNs and convert continuous gradients into spike-compatible ones using G2S. They handle all-zero gradients with RSF, flipping spike states probabilistically while controlling perturbation.

Strengths: The attack achieves high success rates with low perturbation and supports both spike and image inputs. It works directly in SNNs without relying on ANN conversion.

Weaknesses: It assumes a white-box setting with full model access and is mostly validated on classification tasks. The success relies on specific training configurations and thresholds.

Challenges and Future Directions: Future work includes extending the approach to black-box attacks and developing real-time defenses for neuromorphic hardware.

Applications: The method is ideal for evaluating and improving the security of neuromorphic SNN-based vision systems.

Course Website: https://github.com/asifulhoque23/utsa-cs5463-course-website

References:

Bagheri, A., Simeone, O., & Rajendran, B. (2018). *Adversarial Training for Probabilistic Spiking Neural Networks* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1802.08567

Bohte, S. M., Joost N Kok, & Han La Poutré. (2000). SpikeProp: Backpropagation for Networks of Spiking Neurons. *In ESANN*, *(Vol. 48*, 419-424). Retrieved from https://api.semanticscholar.org/CorpusID:14069916

Buzsáki, G. (2006). *Rhythms of the Brain*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195301069.001.0001

Cruz-Albrecht, J. M., Yung, M. W., & Srinivasa, N. (2012). Energy-Efficient Neuron, Synapse and STDP Integrated Circuits. *IEEE Transactions on Biomedical Circuits and Systems*, *6*(3), 246–256. https://doi.org/10.1109/TBCAS.2011.2174152

Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, *9*, 99. https://doi.org/10.3389/fncom.2015.00099

Ghosh-Dastidar, S., & Adeli, H. (2009). SPIKING NEURAL NETWORKS. *International Journal of Neural Systems*, *19*(04), 295–308. https://doi.org/10.1142/S0129065709002002

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, *117*(4), 500–544. https://doi.org/10.1113/jphysiol.1952.sp004764

Hunsberger, E., & Eliasmith, C. (2015). *Spiking Deep Networks with LIF Neurons*. arXiv preprint arXiv: 1510.08829. https://doi.org/10.48550/ARXIV.1510.08829

Iaboni, C., & Abichandani, P. (2024). Event-based Spiking Neural Networks for Object Detection: Datasets, Architectures, Learning Rules, and Implementation. *IEEE Access*, *12*, 180532–180596. https://doi.org/10.1109/ACCESS.2024.3479968

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning*, 448–456. https://doi.org/10.48550/ARXIV.1502.03167

Jin, Y., Zhang, W., & Li, P. (2018). Hybrid Macro/Micro Level Backpropagation for Training Deep Spiking Neural Networks. *Advances in Neural Information Processing Systems*, *31*. https://doi.org/10.48550/ARXIV.1805.07866

Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization* (Version 9). arXiv preprint arXiv:1412.6980. https://doi.org/10.48550/ARXIV.1412.6980

Kudithipudi, D., Schuman, C., Vineyard, C. M., Pandit, T., Merkel, C., Kubendran, R., Aimone, J. B., Orchard, G., Mayr, C., Benosman, R., Hays, J., Young, C., Bartolozzi, C., Majumdar, A., Cardwell, S. G., Payvand, M., Buckley, S., Kulkarni, S., Gonzalez, H. A., … Furber, S. (2025). Neuromorphic computing at scale. *Nature, 637*(8047), 801–812. https://doi.org/10.1038/s41586-024-08253-8

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Lee, J. H., Delbruck, T., & Pfeiffer, M. (2016). Training Deep Spiking Neural Networks Using Backpropagation. *Frontiers in Neuroscience*, *10*, 508. https://doi.org/10.3389/fnins.2016.00508

Legenstein, R., Naeger, C., & Maass, W. (2005). What Can a Neuron Learn with Spike-Timing-Dependent Plasticity? *Neural Computation*, *17*(11), 2337–2382. https://doi.org/10.1162/0899766054796888

Liang, L., Hu, X., Deng, L., Wu, Y., Li, G., Ding, Y., Li, P., & Xie, Y. (2020). *Exploring Adversarial Attack in Spiking Neural Networks with Spike-Compatible Gradient* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2001.01587

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, *10*(9), 1659–1671. https://doi.org/10.1016/S0893-6080(97)00011-7

Markram, H., Gerstner, W., & Sjöström, P. J. (2012). Spike-Timing-Dependent Plasticity: A Comprehensive Overview. *Frontiers in Synaptic Neuroscience*, *4*, 2. https://doi.org/10.3389/fnsyn.2012.00002

Masquelier, T., & Thorpe, S. J. (2007). Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity. *PLoS Computational Biology*, *3*(2), e31. https://doi.org/10.1371/journal.pcbi.0030031

McKennoch, S., Dingding Liu, & Bushnell, L. G. (2006). Fast Modifications of the SpikeProp Algorithm. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 3970–3977. https://doi.org/10.1109/IJCNN.2006.246918

Neftci, E. O., Mostafa, H., & Zenke, F. (2019). Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine*, *36*(6), 51–63. https://doi.org/10.1109/MSP.2019.2931595

Niu, L.-Y., Wei, Y., Liu, W.-B., Long, J.-Y., & Xue, T. (2023). Research Progress of spiking neural network in image classification: A review. *Applied Intelligence*, *53*(16), 19466–19490. https://doi.org/10.1007/s10489-023-04553-0

Pfeiffer, M., & Pfeil, T. (2018). Deep Learning With Spiking Neurons: Opportunities and Challenges. *Frontiers in Neuroscience*, *12*, 774. https://doi.org/10.3389/fnins.2018.00774

Pietrzak, P., Szczęsny, S., Huderek, D., & Przyborowski, Ł. (2023). Overview of Spiking Neural Network Learning Approaches and Their Computational Complexities. *Sensors*, *23*(6), 3037. https://doi.org/10.3390/s23063037

Rathi, N., Chakraborty, I., Kosta, A., Sengupta, A., Ankit, A., Panda, P., & Roy, K. (2023). Exploring Neuromorphic Computing Based on Spiking Neural Networks: Algorithms to Hardware. *ACM Computing Surveys*, *55*(12), 1–49. https://doi.org/10.1145/3571155

Rathi, N., Srinivasan, G., Panda, P., & Roy, K. (2020). *Enabling Deep Spiking Neural Networks with Hybrid Conversion and Spike Timing Dependent Backpropagation* (Version 1). arXiv preprint arXiv:2005.01807. https://doi.org/10.48550/ARXIV.2005.01807

Rueckauer, B., & Liu, S.-C. (2018). Conversion of analog to spiking neural networks using sparse temporal coding. *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5. https://doi.org/10.1109/ISCAS.2018.8351295

Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M., & Liu, S.-C. (2017). Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification. *Frontiers in Neuroscience*, *11*, 682. https://doi.org/10.3389/fnins.2017.00682

Ruf, B., & Schmitt, M. (1997). Learning temporally encoded patterns in networks of spiking neurons. *Neural Processing Letters*, *5*(1), 9–18. https://doi.org/10.1023/A:1009697008681

Sanaullah, Koravuna, S., Rückert, U., & Jungeblut, T. (2023). Exploring spiking neural networks: A comprehensive analysis of mathematical models and applications. *Frontiers in Computational Neuroscience*, *17*, 1215824. https://doi.org/10.3389/fncom.2023.1215824

Sengupta, A., Ye, Y., Wang, R., Liu, C., & Roy, K. (2019). Going Deeper in Spiking Neural Networks: VGG and Residual Architectures. *Frontiers in Neuroscience*, *13*, 95. https://doi.org/10.3389/fnins.2019.00095

Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition* (Version 6). arXiv preprint arXiv:1409.1556. https://doi.org/10.48550/ARXIV.1409.1556

Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, *3*(9), 919–926. https://doi.org/10.1038/78829

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.48550/ARXIV.1512.00567

Tan, C., Šarlija, M., & Kasabov, N. (2020). Spiking Neural Networks: Background, Recent Development and the NeuCube Architecture. *Neural Processing Letters*, *52*(2), 1675–1701. https://doi.org/10.1007/s11063-020-10322-8

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2019). Deep learning in spiking neural networks. *Neural Networks*, *111*, 47–63. https://doi.org/10.1016/j.neunet.2018.12.002

Thorpe, S., & Gautrais, J. (1998). Rank Order Coding. In J. M. Bower (Ed.), *Computational Neuroscience* (pp. 113–118). Springer US. https://doi.org/10.1007/978-1-4615-4831-7_19

Wang, X., Lin, X., & Dang, X. (2020). Supervised learning in spiking neural networks: A review of algorithms and evaluations. *Neural Networks*, *125*, 258–280. https://doi.org/10.1016/j.neunet.2020.02.011

Wu, H., Zhang, Y., Weng, W., Zhang, Y., Xiong, Z., Zha, Z.-J., Sun, X., & Wu, F. (2021). Training Spiking Neural Networks with Accumulated Spiking Flow. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 10320–10328. https://doi.org/10.1609/aaai.v35i12.17236

Wu, J., Chua, Y., Zhang, M., Yang, Q., Li, G., & Li, H. (2019). Deep Spiking Neural Network with Spike Count based Learning Rule. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–6. https://doi.org/10.1109/IJCNN.2019.8852380

Wu, Y., Deng, L., Li, G., Zhu, J., & Shi, L. (2018). Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks. *Frontiers in Neuroscience*, *12*, 331. https://doi.org/10.3389/fnins.2018.00331

Xu, Y., Zeng, X., & Zhong, S. (2013). A New Supervised Learning Algorithm for Spiking Neurons. *Neural Computation*, *25*(6), 1472–1511. https://doi.org/10.1162/NECO_a_00450

Yamazaki, K., Vo-Ho, V.-K., Bulsara, D., & Le, N. (2022). Spiking Neural Networks and Their Applications: A Review. *Brain Sciences*, *12*(7), 863. https://doi.org/10.3390/brainsci12070863

Zenke, F., & Ganguli, S. (2018). SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks. *Neural Computation*, *30*(6), 1514–1541. https://doi.org/10.1162/neco_a_01086

Zhang, L., Zhou, S., Zhi, T., Du, Z., & Chen, Y. (2019). TDSNN: From Deep Neural Networks to Deep Spike Neural Networks with Temporal-Coding. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 1319–1326. https://doi.org/10.1609/aaai.v33i01.33011319