Neuromorphic Sentiment Analysis Using Spiking Neural Networks

Subjects: Computer Science, Artificial Intelligence

Contributor: Raghavendra K. Chunduri , Darshika G. Perera

Spiking neural networks, often employed to bridge the gap between machine learning and neuroscience fields, are considered a promising solution for resource-constrained applications. Since deploying spiking neural networks on traditional von-Newman architectures requires significant processing time and high power, typically, neuromorphic hardware is created to execute spiking neural networks. The objective of neuromorphic devices is to mimic the distinctive functionalities of the human brain in terms of energy efficiency, computational power, and robust learning.

neuromorphic computing | artificial neural network | natural language processing

1. Introduction

In recent years, the Artificial Neural Networks (ANNs) domain has witnessed a significant adaptation of Deep Neural Networks (DNNs) across several fields, such as machine learning, computer vision, artificial intelligence, and natural language processing (NLP). DNNs are capable of accurately performing a wide range of tasks by training on massive datasets [1]. However, the energy consumption and computational cost required for training large volumes of datasets and for deploying the resulting applications have been of less importance; thus, they have been overlooked [2][3]. The DNNs typically consume high power and require large data storage [4][5]. Although there have been significant advancements in ANNs, ANNs were unable to achieve the same level of energy efficiency and online learning ability as biological neural networks [6]. Drawing inspiration from brain-inspired computing, one potential solution to address the issue of high-power consumption is to use the neuromorphic hardware with Spiking Neural Networks (SNNs). SNNs, often considered the third generation of neural networks, are emerging to bridge the gap between fields such as machine learning and neuroscience [7].

Unlike traditional neural networks that rely on continuous-valued signals, the SNNs work in continuous time [8]. In SNNs, the neurons communicate with each other using discrete electrical signals called spikes. Spikes model the behavior of the neurons more accurately and more biologically plausible than ANNs, thus making SNNs more energy efficient and computationally powerful than ANNs [9]. The neuron models of ANNs and SNNs differ from each other. For instance, ANNs do not have any memory and use sigmoid, tanh, or rectified linear unit (ReLU) as computational units, whereas SNNs have memory and use non-differentiable neuron models. Typically, large-scale SNN models consume high power and require high execution time when utilized/executed on classical Von Neumann architectures [10]. Hence, there is a need for high-speed and low-power hardware for executing large-scale SNN models. In this regard, existing neuromorphic platforms, such as SpiNNaker [11], Loihi [12], NeuroGrid

[13], and TrueNorth by IBM [14], are expected to advance the applicability of large-scale SNNs in several emerging fields by offering energy-efficient high-speed computational solutions. SNNs have the functional similarities to biological neural networks, allowing them to embrace the sparsity and temporal coding found in biological systems [15]. However, SNNs are difficult to train because of their non-differentiable neuron models. In terms of speed performance, SNNs are inferior to DNNs. Nevertheless, due to the low power traits, SNNs are considered more efficient than DNNs [6].

2. Neuromorphic Sentiment Analysis Using Spiking Neural Networks

2.1. Spiking Neural Networks (SNNs)

As stated in [16], the SNNs are considered the third generation of neural networks, which communicate through a sequence of discrete electrical events called “spikes” that takes place at a point of time. The SNN models are generally expressed in the form of differential equations [17]. The structure of spiking neurons in the SNN model is similar to the structure of the ANN neuron; however, their behavior is different. SNNs are widely used in various applications, including brain–machine interface, event detection, forecasting, and decision making [18][19].

The spiking neuron models are distinguished based on the biological plausibility and computational capabilities [20][21][22]. Typically, spiking neuron models are selected based on specific user requirements. **Figure 1** illustrates the schematics of a biological neural network, ANN and SNN [17].

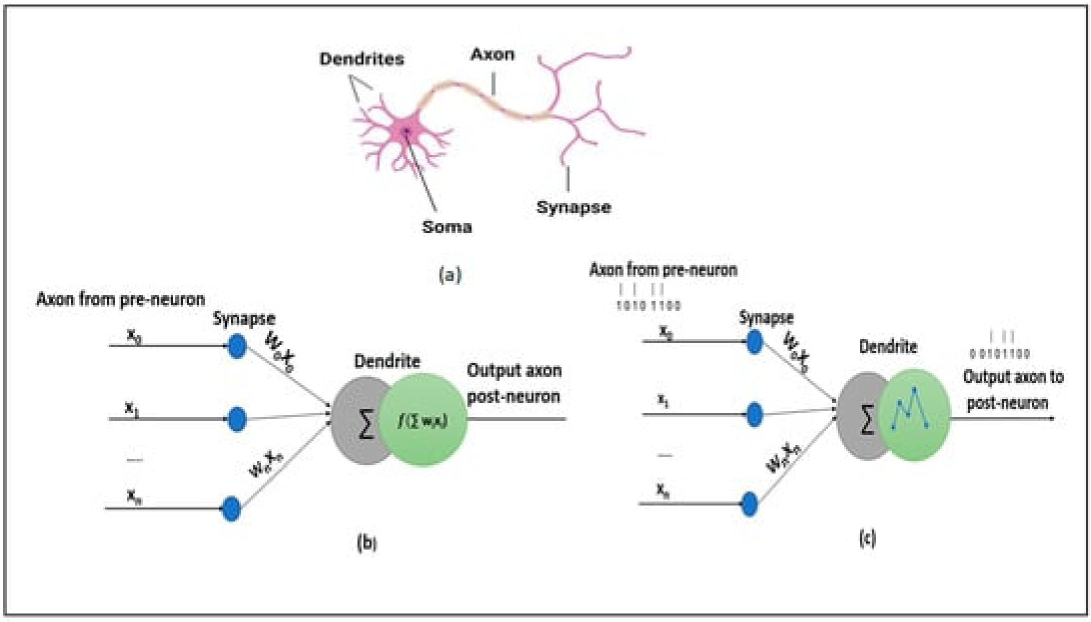

Figure 1. Schematic representations of (**a**) biological neural network, (**b**) artificial neural network, and (**c**) spiking neural network.

2.2. ANN to SNN Conversion

The ANNs are used extensively for solving several tasks in various fields, such as machine learning and artificial intelligence. In this case, deep learning develops large neural networks with millions of neurons that span up to thousands of layers. These large neural networks have proven to be effective while solving several complex tasks, including video classification, object detection and recognition, etc.; however, these networks require massive computational resources [23][24][25][26]. The development of SNNs is mainly to address the challenge associated with massive computational resources. The SNNs perform similar tasks with less computational resources and with low energy consumption. In SNNs, all the computations are event-driven, and operations are sparse. In this case, the computations and operations are performed only when there is a significant change in the input. Typically, training a large SNN is a difficult task; thus, an alternative approach is to take a pre-trained ANN network and convert it into SNNs [1]. Existing ANN-to-SNN conversion methods in the literature primarily focus on converting ReLu to IF neurons.

An overview of ANN-to-SNN conversion is illustrated in **Figure 2**. The process of converting from ANN to SNN involves transferring the trained ANN settings that use ReLU activations to an SNN with an identical structure, as depicted in **Figure 2**. This approach enables the SNN to achieve exceptional performance while requiring minimal computational resources. Initially, the ANN model is trained with the given inputs, and the weights are saved. Typically, traditional trained ANN models are being executed on GPUs, as illustrated in top modules in blue (in **Figure 2**).

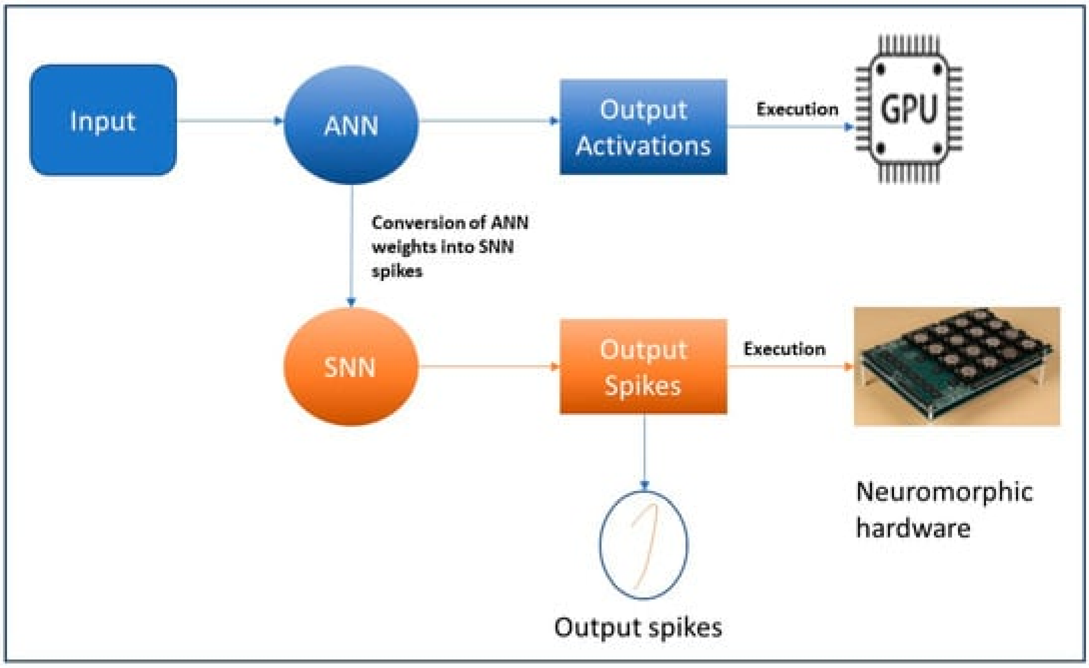

Figure 2. Overview of the ANN-to-SNN conversion.

2.3. Neuromorphic Hardware

The neuromorphic hardware for SNNs is categorized into analog, digital, or mixed-signal (analog/digital) designs [27]. Many neuromorphic hardware platforms with varying configurations have emerged to manage large-scale neural networks. From these neuromorphic platforms, fully digital and mixed-signal hardware, such as IBM TrueNorth, NeuroGrid, BrainScaleS, Lohi, and SpiNNaker, are some of the commonly used platforms among several applications [28]. A detailed description of the neuromorphic hardware platforms can be found in [28].

Table 3 presents various features/characteristics of existing neuromorphic hardware platforms

Table 3. Characteristics of existing neuromorphic hardware platforms.

Platform	Technology (mm)	Electronics	Chip Area (mm^2)	Neuron Model	On-Chip Learning	Neuron Number (Chip)	Synapse Model	Synapse Number (Chip)	Online Learning	Power
TrueNorth [9]	ASIC-CMOS 28	Digital	430	LIF	No	1 Million	Binary 4 modulators	256 M	No	65 mW (per chip)
BrainScaleS [9]	ASIC-CMOS 180	Analog/Digital	50	Adaptive exponential IF	No	512	Spiking 4-bit digital	100 K	Yes	2 kW Per module (peak)
NeuroGrid [9]	ASIC-CMOS 180	Analog/Digital	168	Adaptive Quadratic IF	No	65,000	Shared dendrite	100 M	Yes	2.7 W
Loihi [9]	ASIC-CMOS 14 nm	Digital	60	LIF	Yes (with plasticity rule)	131,000	N/A	126 M	Yes	0.45 W
SpiNNaker [9]	ASIC-CMOS 130 nm	Digital	102	LIF LZH HH	Yes (synaptic plasticity rule)	16,000	Programmable	16 M	Yes	1 W (per chip)

2.4. SpiNNaker

The SpiNNaker was designed by the Advanced Processor Technologies Research Group (APT), from the Department of Computer Science at the University of Manchester [29]. It is composed of 57,600 processing nodes, each with 18 ARM9 processors (specifically ARM968), 128 MB of mobile DDR-SDRAMs, totaling 1,036,800 cores, and over 7 TB of RAM [30][31]. The SpiNNaker is an SNN architecture designed to simulate large-scale SNNs. The main component of the SpiNNaker system is the SpiNNaker chip, whose main focus is to provide the required scalability and flexibility to perform experiments with neuron models. Based on brain-inspired computing, the objective of the SpiNNaker system is to design the neural architecture model of the human brain which is made up of approximately 100 billion neurons connected by trillions of synapses [29]. The SpiNNaker machine is a collection of low-power processors, which can simulate/execute a small number of neurons and synapses in real time. In this case, all the processors are interconnected by a high-speed network [32]. The high-speed network allows the

References

1. Rueckauer, B.; Lungu, I.A.; Hu, Y.; Pfeiffer, M.; Liu, S.C. Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification. Front. Neurosci. 2017, 11, 682.
2. Soman, S.; jayadeva; Suri, M. Recent trends in neuromorphic engineering. Big Data Anal. 2016, 1, 15.
3. Chandarana, P.; Mohammadi, M.; Seekings, J.; Zand, R. Energy-Efficient Deployment of Machine Learning Workloads on Neuromorphic Hardware. In Proceedings of the 2022 IEEE13th International Green and Sustainable Computing Conference (IGSC), Los Alamitos, CA, USA, 24–25 October 2022; pp. 1–7.
4. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. In Proceedings of the 4th International

processors to communicate with each other, while distributing the computation load for simulating a large neural network. The main advantage of the SpiNNaker system is its ability to simulate large-scale neural networks using an asynchronous scheme of communication [30][33], which is essential for testing brain functions and developing new neural network applications in areas such as robotics, machine learning, and artificial intelligence. The SpiNNaker system is indeed an exciting creation in the field of neural networks, and it has the potential to greatly advance the understanding of the brain and the information processing of the brain [34][35].

2.4.1. Architecture of SpiNNaker Chip

As stated in [35], the SpiNNaker chip has 18 cores coupled with an external RAM controller and a Network-on-Chip (NoC). Each core comprises an ARM968 processor, a direct memory access controller, a controller for communications, a network interface controller, and other peripherals, including a timer [35]. Every core in the SpiNNaker chip runs given applications by simulating/executing a group of neurons at 200 MHz. Each core also comprises 96 kB of tightly coupled memory (TCM). In order to avoid any contention issues, this TCM is split into two: 64 kB for data (DTCM), and 32 kB for instructions (ITCM). The DTCM consists of application data, including zero-initialized data, heap, stack, and read/write. Each chip in the SpiNNaker system has 128 MB of shared memory (i.e., SDRAM), which is directly accessible by all cores in the SpiNNaker chip [35]. In this case, the memory access time varies significantly when accessing the different memories mentioned above. Hence, the following should be considered when designing applications for the SpiNNaker system.

- Faster access to DTCM at ≈5 ns/word. DTCM is only limited to the local core.
- Access to SDRAM via a bridge. Accessing SDRAM could lead to a contention issue, since more than one core in the SpiNNaker chip could attempt to access. This is a slow process with >100 ns/word.
- As a result, each core encompasses a direct memory access (DMA) controller, which is used to enable bulk transfer of data from the SDRAM core to DTCM efficiently. Although the DMCA setup introduces a fixed overhead, the data are still transferred from the processor independently at ≈10 ns/word.

The SpiNNaker is a large-scale parallel network, comprising low-power and energy-efficient processors connected by a network. Each node in the network is responsible for simulating/executing a small number of neurons and synapses [34]. Each node in the network communicates with every other node to exchange information and distribute the computation load. Each node in the network consists of processors, memory, I/O interfaces and core. Every node in the SpiNNaker architecture is constructed from one or more SpiNNaker boards, which are made up of SpiNNaker chips [11]. Currently, two production versions called SpiNN-3 and SpiNN-5, each of which has 4 and 48 chips, respectively, are available.

2.4.2. Components of SpiNNaker System

The architecture of the SpiNNaker system consists of the following four major components [11].

Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016; Conference Track Proceedings; Bengio, Y., LeCun, Y., Eds.; ICLR: San Juan, Puerto Rico, 2016.

5. Strubell, E.; Ganesh, A.; McCallum, A. Energy and Policy Considerations for Modern Deep Learning Research. Proc. AAAI Conf. Artif. Intell. 2020, 34, 13693–13696.
6. Deng, L.; Wu, Y.; Liang, L.; Ding, Y.; Li, G.; Zhao, G.; Li, P.; Xie, Y. Rethinking the performance comparison between SNNS and ANNS. Neural Netw. 2019, 121, 294–307.
7. Diehl, P.U.; Cook, M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Front. Comput. Neurosci. 2015, 9, 99.
8. Jang, H.; Simeone, O.; Gardner, B.; Gruning, A. An Introduction to Probabilistic Spiking Neural Networks: Probabilistic Models, Learning Rules, and Applications. IEEE Signal Process. Mag. 2019, 36, 64–77.
9. Javanshir, A.; Nguyen, T.T.; Mahmud, M.A.P.; Kouzani, A.Z. Advancements in Algorithms and Neuromorphic Hardware for Spiking Neural Networks. Neural Comput. 2022, 34, 1289–1328.
10. Furber, S.B.; Galluppi, F.; Temple, S.; Plana, L.A. The SpiNNaker Project. Proc. IEEE 2014, 102, 652–665.
11. Furber, S.B.; Lester, D.R.; Plana, L.A.; Garside, J.D.; Painkras, E.; Temple, S.; Brown, A.D. Overview of the SpiNNaker System Architecture. IEEE Trans. Comput. 2013, 62, 2454–2467.
12. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. IEEE Micro 2018, 38, 82–99.
13. Benjamin, B.V.; Gao, P.; McQuinn, E.; Choudhary, S.; Chandrasekaran, A.R.; Bussat, J.M.; Alvarez-Icaza, R.; Arthur, J.V.; Merolla, P.A.; Boahen, K. Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations. Proc. IEEE 2014, 102, 699–716.
14. Diehl, P.U.; Pedroni, B.U.; Cassidy, A.; Merolla, P.; Neftci, E.; Zarrella, G. TrueHappiness: Neuromorphic emotion recognition on TrueNorth. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 4278–4285.
15. Kasabov, N.K. Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence; Springer: Berlin/Heidelberg, Germany, 2019; pp. 431–456.
16. Maass, W. Networks of spiking neurons: The third generation of neural network models. Neural Netw. 1997, 10, 1659–1671.
17. Yamazaki, K.; Vo-Ho, V.K.; Bulsara, D.; Le, N. Spiking Neural Networks and Their Applications: A Review. Brain Sci. 2022, 12, 863.

Processing nodes: are the individual processors used to simulate/execute the behavior of artificial neurons and synapses.

Interconnect fabric: is a high-speed network used to connect the processing nodes together. Interconnect fabric allows efficient communication between the nodes, as well as efficient load distribution across the network.

Host machine: is a separate master computer/processor used to configure and control the SpiNNaker system. The host machine constantly communicates with the processing nodes via the network interface. The host machine also provides a user interface to set up and run the simulations.

Software stack: consists of a variety of software components that work together to enable the simulation/execution of neural networks on the SpiNNaker system. This software stack includes the operating system running on the processing nodes, higher-level software libraries, and tools for configuring and running the simulations.

2.5. PyNN

In [36], the authors introduced PyNN, which is a python interface used to define the simulations after creating the SNN model. The simulations are typically executed on the SpiNNaker machine via an event-driven operating system [36]. Using a python script, PyNN allows users to specify the SNN simulations for executions. In this case, NEST, Neuron, INI, Brian, and SpiNNaker are commonly used SNN simulators.

2.6. Sentiment Analysis Using Natural Language Processing

Sentiment analysis is a natural language processing (NLP) technique that is commonly used to identify, extract, and quantify subjective information from text data [37]. Sentiment analysis is mainly used to analyze the text and determine the sentiment score. The sentiment score can range from −1 (indicating very negative sentiment) to +1 (indicating very positive sentiment), with 0 representing the neutral sentiment [38]. Using deep-learning-based approaches to perform sentiment analysis in NLP is a popular research area. Sentiment analysis is widely employed across various fields, such as marketing, finance, and customer service, to name a few [39]. Sentiment analysis can also be used to analyze financial news and social media to predict stock prices or market trends [40]. However, with the ongoing increase in data sizes, novel and efficient models (for sentiment analysis) are needed to manage and process the massive amount of data [41]. Although the existing ANN models provide the required accuracy while classifying the data, the ANN models are not efficient in terms of energy consumption and speed-performance [42].

18. Mead, C. Neuromorphic electronic systems. Proc. IEEE 1990, 78, 1629–1636.
19. Calimera, A.; Macii, E.; Poncino, M. The Human Brain Project and neuromorphic computing. Funct. Neurol. 2013, 28, 191–196.
20. Izhikevich, E. Which model to use for cortical spiking neurons? IEEE Trans. Neural Netw. 2004, 15, 1063–1070.
21. Abusnaina, A.; Abdullah, R. Spiking Neuron Models: A Review. Int. J. Digit. Content Technol. Its Appl. 2014, 8, 14–21.
22. Amunts, K.; Ebell, C.; Muller, J.; Telefont, M.; Knoll, A.; Lippert, T. The Human Brain Project: Creating a European Research Infrastructure to Decode the Human Brain. Neuron 2016, 92, 574–581.
23. Rueckauer, B.; Liu, S.C. Conversion of analog to spiking neural networks using sparse temporal coding. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
24. Diehl, P.U.; Neil, D.; Binas, J.; Cook, M.; Liu, S.C.; Pfeiffer, M. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–8.
25. Sengupta, A.; Ye, Y.; Wang, R.; Liu, C.; Roy, K. Going deeper in spiking neural networks: VGG and residual architectures. Front. Neurosci. 2019, 13, 95.
26. Patel, K.; Hunsberger, E.; Batir, S.; Eliasmith, C. A spiking neural network for image segmentation. arXiv 2021, arXiv:2106.08921.
27. Neil, D.; Liu, S.C. Effective sensor fusion with event-based sensors and deep network architectures. In Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, 22–25 May 2016; pp. 2282–2285.
28. Camuñas-Mesa, L.A.; Linares-Barranco, B.; Serrano-Gotarredona, T. Neuromorphic Spiking Neural Networks and Their Memristor-CMOS Hardware Implementations. Materials 2019, 12, 2745.
29. Fonseca Guerra, G.A.; Furber, S.B. Using Stochastic Spiking Neural Networks on SpiNNaker to Solve Constraint Satisfaction Problems. Front. Neurosci. 2017, 11, 714.
30. Furber S Large-scale neuromorphic computing systems. J. Neural Eng. 2016, 13, 051001.
31. Jin, X.; Galluppi, F.; Patterson, C.; Rast, A.; Davies, S.; Temple, S.; Furber, S. Algorithm and software for simulation of spiking neural networks on the multi-chip spinnaker system. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 12–23 July 2010; pp. 1–8.

32. Brown, A.D.; Furber, S.B.; Reeve, J.S.; Garside, J.D.; Dugan, K.J.; Plana, L.A.; Temple, S. Spinnaker—Programming model. IEEE Trans. Comput. 2015, 64, 1769–1782.

33. Rowley, A.G.D.; Brenninkmeijer, C.; Davidson, S.; Fellows, D.; Gait, A.; Lester, D.R.; Plana, L.A.; Rhodes, O.; Stokes, A.B.; Furber, S.B. Spinntools: The execution engine for the spinnaker platform. Front. Neurosci. 2019, 13, 231. Available online: https://www.frontiersin.org/articles/10.3389/fnins.2019.00231 (accessed on 20 May 2023).

34. Sen-Bhattacharya, B.; James, S.; Rhodes, O.; Sugiarto, I.; Rowley, A.; Stokes, A.B.; Gurney, K.; Furber, S.B. Building a spiking neural network model of the basal ganglia on spinnaker. IEEE Trans. Cogn. Dev. Syst. 2018, 10, 823–836.

35. Rhodes, O.; Bogdan, P.A.; Brenninkmeijer, C.; Davidson, S.; Fellows, D.; Gait, A.; Lester, D.R.; Mikaitis, M.; Plana, L.A.; Rowley, A.G.D.; et al. Spynnaker: A software package for running pynn simulations on spinnaker. Front. Neurosci. 2018, 12, 816. Available online: https://www.frontiersin.org/articles/10.3389/fnins.2018.00816 (accessed on 20 May 2023).

36. Davison, A.; Bru¨derle, D.; Eppler, J.; Kremkow, J.; Muller, E.; Pecevski, D.; Perrinet, L.; Yger, P. Pynn: A common interface for neuronal network simulators. Front. Neurosci. 2009, 2, 11. Available online: https://www.frontiersin.org/articles/10.3389/neuro.11.011.2008 (accessed on 20 May 2023).

37. Nagarhalli, T.P.; Mhatre, S.; Patil, S.; Patil, P. The review of natural language processing applications with emphasis on machine learning implementations. In Proceedings of the 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 6–8 March 2022; pp. 1353–1358.

38. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment analysis based on deep learning: A comparative study. Electronics 2020, 9, 483.

39. Shafin, M.A.; Hasan, M.M.; Alam, M.R.; Mithu, M.A.; Nur, A.U.; Faruk, M.O. Product review sentiment analysis by using nlp and machine learning in bangla language. In Proceedings of the 2020 23rd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 19–21 December 2020; pp. 1–5.

40. Chen, C.-H.; Chen, P.-Y.; Lin, J.C.-W. An Ensemble Classifier for Stock Trend Prediction Using Sentence-Level Chinese News Sentiment and Technical Indicators. Int. J. Interact. Multimed. Artif. Intell. 2022, 7, 53–64.

41. Chunduri, R.K.; Cherukuri, A.K. Big Data Processing Frameworks and Architectures: A Survey in Hand Book of Big Data Analytics; IET Digital Library: Stevenage, UK, 2021; Volume 1, pp. 37–104. Available online: https://digital-library.theiet.org/content/books/10.1049/pbpc037fch2 (accessed on 15 May 2023).

42. Ricketts, J.; Barry, D.; Guo, W.; Pelham, J. A scoping literature review of natural language processing application to safety occurrence reports. Safety 2023, 9, 22.