MATH 456: Final Project Report

Bridging Biological Neurons and Artificial Activations: Feed-Forward and Spiking Neural Networks via the Leaky Integrate-and-Fire Model

Helektra Katsoulakis[1]

[1]University of Massachusetts Amherst

December 17, 2025

Abstract

Biological neural systems perform reliable and flexible computation under strict energy constraints. This efficiency motivates neuromorphic approaches that move away from dense, continuous activations toward sparse, event-driven spikes. The leaky integrate-and-fire (LIF) neuron offers a simple model of membrane voltage dynamics and spike generation, making it well suited for analyzing subthreshold integration, repetitive spiking, and firing-rate versus input-current (F–I) behavior [3].

In this work, LIF dynamics are simulated and non-dimensionalized to show a universal, parameter-independent form that highlights the key factors that control when a neuron spikes. These dynamics are then compared with common feed-forward neural network (FFNN) activation functions to assess which biological behaviors are captured by ReLU and sigmoid mappings [5]. At the network level, an FFNN and a spiking neural network (SNN) constructed from LIF neurons are evaluated on MNIST digit classification. While achieving comparable classification accuracy, the SNN requires substantially fewer operations due to sparse spike activity [4, 5]. Together, these results support event-driven computation as a promising direction toward more energy-efficient artificial intelligence [1].

Keywords: leaky integrate-and-fire, spiking neural networks, surrogate gradients, neuromorphic computing, MNIST

1 Introduction

Modern deep learning achieves strong performance across a wide range of tasks but they typically rely on dense, continuous-valued computation. In these models, nearly all neurons evaluate weighted sums and nonlinearities at every forward pass, leading to high computational and energy costs [1]. Training large-scale models can therefore require enormous amounts of power; for example, training GPT-3 has been estimated to consume on the order of 190,000 kilowatt-hours of energy. In contrast, the human brain operates on roughly 20 W of power, which is about 0.48 kWh per day [5]. This contrast suggests that biological and artificial systems approach computation very differently.

That difference lies not only in scale, but in computational structure. Artificial neural networks rely on dense, continuous calculations, where nearly every neuron computes on every input. Biological neurons operate very differently. Information is communicated through brief electrical spikes, and most neurons remain silent a majority of the time. This sparse, event-driven signaling allows computation to occur only when necessary, avoiding much of the unnecessary activity seen in artificial networks [1]. Biological systems support reliable and flexible information processing while consuming little power.

A starting point for exploring this question is the leaky integrate-and-fire (LIF) neuron model. Originating from early quantitative models of neural excitation and later adopted as a standard reduced model in computational neuroscience [3], the LIF ODE captures three essential features of neuronal behavior: integration of incoming input, leakage toward a resting potential, and spike emission once a threshold is reached, followed by reset and refractory dynamics. Although highly simplified, the LIF model produces a sharp activation threshold and firing-rate saturation at high input levels due to reset and refractory constraints.

These properties make the LIF neuron a useful reference for comparing biological excitability to standard artificial activation functions. In particular, the model helps clarify why standard feed-forward activations only partially resemble biological input–output behavior.

This project investigates two related questions: (1) how simulated LIF firing-rate behavior compares to common feed-forward neural network (FFNN) activations such as ReLU and sigmoid [5], and (2) whether a spiking neural network composed of LIF neurons can achieve classification accuracy comparable to a matched FFNN [4, 5].

2 Materials & Methods

2.1 LIF neuron model and simulation setup

The leaky integrate-and-fire (LIF) model provides a simple mathematical description of how a biological neuron responds to incoming electrical input. In this model, the neuron's membrane is treated as a passive RC circuit that integrates input current while continuously leaking charge back toward a resting potential [3]. Let $V(t)$ denote the membrane potential. The LIF ODE is represented by

$$C_m \frac{dV}{dt} = -g_L (V - E_L) + I_{\text{in}}, \tag{1}$$

where C_m is the membrane capacitance, g_L is the leak conductance, E_L is the resting (leak reversal) potential, and I_{in} represents the input current [3].

To simulate this model, the LIF equation is evaluated using a fixed set of biophysically reasonable parameters. Model parameters were chosen to be consistent with standard values used in computational neuroscience. Time constants were set on the order of tens of milliseconds, and voltage thresholds were chosen tens of millivolts above rest [3]. These choices preserve biologically realistic behavior while keeping the model simple enough for numerical simulation and analysis. Table 1 summarizes the parameters used throughout this work.

For convenience, the equation is rewritten in terms of the membrane time constant $\tau_m = C_m/g_L$, which yields the equivalent form

$$\tau_m \frac{dV}{dt} = -(V - E_L) + \frac{I_{\text{in}}}{g_L}. \tag{2}$$

Rewriting in this form makes the characteristic time scale of the neuron explicit and clarifies how quickly the membrane voltage responds to input.

Table 1: Leaky Integrate-and-Fire (LIF) neuron parameters used in simulations.

Parameter	Value	Description
τ_m	10 ms	Membrane time constant controlling voltage response speed.
g_L	10 nS	Leak conductance setting passive decay toward rest.
C_m	100 pF	Membrane capacitance, with $C_m = \tau_m g_L$.
E_L	-75 mV	Resting (leak reversal) potential.
V_{th}	-55 mV	Spike threshold voltage.
V_{reset}	-75 mV	Post-spike reset voltage.
τ_{ref}	2 ms	Refractory period following a spike.

2.2 Rheobase derivation

An important quantity for understanding LIF behavior is the rheobase current, defined as the smallest constant input current that produces sustained spiking [3]. To derive this value, consider the steady-state solution of (2) under constant input. Setting $dV/dt = 0$ gives

$$0 = -(V_{ss} - E_L) + \frac{I}{g_L} \quad \Rightarrow \quad V_{ss} = E_L + \frac{I}{g_L}. \tag{3}$$

Spiking occurs only if the steady-state voltage reaches or exceeds the threshold V_{th}. Enforcing $V_{ss} \geq V_{\text{th}}$ yields the rheobase current

$$I_{\text{rheo}} = g_L (V_{\text{th}} - E_L). \tag{4}$$

Using the parameter values in Table 1 ($g_L = 10$ nS, $V_{\text{th}} = -55$ mV, and $E_L = -75$ mV), the rheobase current evaluates to

$$I_{\text{rheo}} = 200 \text{ pA}.$$

This value marks a significant transition in behavior: inputs below rheobase lead to subthreshold dynamics with no spikes, while inputs above rheobase produce regular firing. This plays a central role in shaping the neuron's input–output relationship.

2.3 F–I curve construction

To quantify the input–output behavior of the LIF neuron, firing-rate versus input-current (F–I) curves were computed. The neuron was simulated numerically for a range of constant input currents using an explicit Euler method. Euler's method was chosen because the LIF model is defined by a first-order ordinary differential equation with a simple threshold-and-reset rule, making it straightforward to implement and sufficient for capturing spike timing at the millisecond scale.

For each input current, spike times were recorded, and the initial transient period was discarded so that firing rates reflected steady behavior. The firing rate $f(I)$ was computed as the total number of spikes divided by the effective simulation time. Refractory dynamics were included explicitly, which limits the maximum firing rate and produces saturation at high input currents [3]. As a result, the F–I curve increases approximately linearly above threshold before flattening at higher currents due to the refractory constraint.

3 Non-Dimensionalization of the LIF Model

To simplify the analysis and make comparisons easier, the LIF model was rewritten in dimensionless form. Starting from

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + I_{\text{in}}, \tag{5}$$

time was scaled by the membrane time constant $\tau_m = C_m/g_L$, voltage was measured relative to the threshold–resting potential difference, and input current was scaled by the rheobase current [3].

With these choices, the dynamics can be written as

$$\frac{d\tilde{v}}{d\tilde{t}} = -\tilde{v} + \tilde{i}, \tag{6}$$

which removes dependence on specific physical units. In this form, behavior is controlled primarily by the dimensionless input $\tilde{i}$. For $\tilde{i} < 1$, the voltage remains below threshold, while $\tilde{i} > 1$ leads to sustained spiking, consistent with the definition of the rheobase current [3].

Rewriting the model this way makes it easier to compare the LIF firing-rate curve with standard artificial activation functions, which are also unitless.

4 Linking Biological Firing Rates to Artificial Activations

4.1 Biological F–I behavior

To show how input current controls neuronal behavior, the LIF model was simulated under several constant input currents, ranging from 150 pA to 350 pA, and spanning subthreshold, near-rheobase, and suprathreshold values. For currents below rheobase, the membrane potential relaxed toward a steady subthreshold value without producing spikes. Near rheobase, the voltage approached threshold but did not fire. Above rheobase, the neuron produced regular spike trains whose frequency increased with input current. These voltage traces show how increasing input current pushes the neuron from subthreshold behavior into sustained spiking. This behavior is summarized by the fitted curve shown in Figure 1

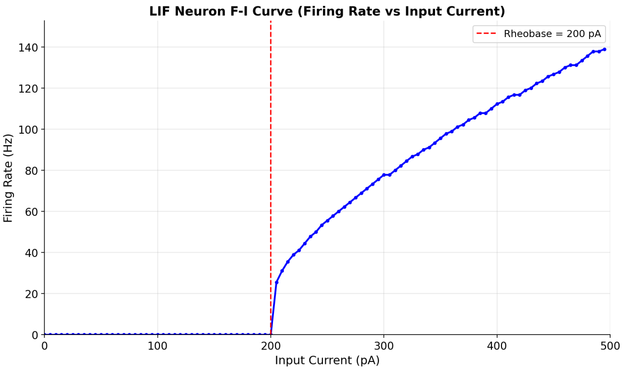

Figure 1: Fitted firing-rate versus input-current (F–I) curve for the LIF neuron.

4.2 F–I curve and activation function comparison

To compare biological firing behavior with artificial activations, ReLU and sigmoid functions were fit to the simulated F–I data. No standard activation function captures both the sharp threshold and saturation behavior simultaneously [5]. This tradeoff is visible in the fitted curves: ReLU captures the abrupt onset of firing but grows without bound, while sigmoid saturates but introduces a smooth, gradual threshold transition.

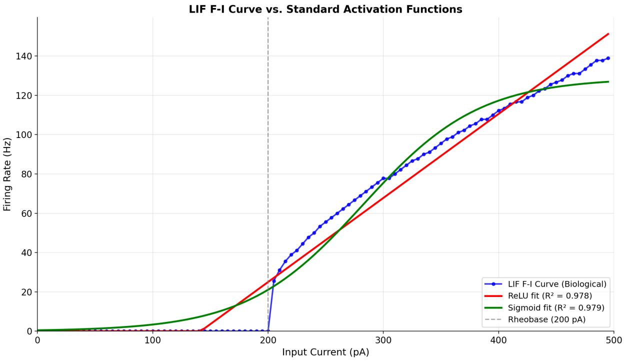

Figure 2: F-I curve in comparison to ReLu and Sigmoid activation functions. The LIF F-I curve is shown in blue, the ReLU fit is shown in red, and Sigmoid in green.

These results show that common activation functions are only approximations of biological neuron behavior and motivate the use of spiking models when biological realism or sparsity is of interest.

5 Network Models

5.1 FFNN baseline (MNIST)

To have a point of comparison, a standard feed-forward neural network is trained on MNIST digit classification. MNIST is chosen because it is simple enough to train quickly, but structured enough to compare accuracy, sparsity, and computational cost across different neuron models. The network uses a fully connected architecture with ReLU activations in the hidden layers and a softmax output layer, trained using cross-entropy loss [4].

This model represents the typical way modern artificial neural networks are implemented. Every neuron computes a weighted sum and activation for every input, regardless of whether that neuron is strongly activated or not. To estimate computational cost, inference was measured using floating-point operation (FLOP) counts based on standard multiply–add calculations for dense layers [1]. Even when a ReLU output is zero, the full computation still occurs, which makes this model a useful reference for dense, continuous-valued computation.

5.2 SNN model (LIF neurons) and surrogate-gradient training

In a similar manner, a spiking neural network using leaky integrate-and-fire neurons is trained on the same MNIST task. The SNN uses the same layer sizes as the FFNN so that both models have the same number of parameters. This allows differences in performance and computational cost to be traced back to the neuron model rather than the network architecture.

In the SNN, neurons communicate through discrete spikes instead of continuous activations. This discrete signaling makes the network event-driven. Neuron states and synaptic updates are only triggered when spikes occur, rather than being recomputed at every forward pass as in dense feed-forward networks. As a result, neurons that remain subthreshold incur no downstream computation. This mirrors biological neural signaling, where neurons remain inactive unless driven above threshold, and stands in contrast to FFNNs where all neurons are evaluated regardless of their contribution.

To train the network, surrogate-gradient methods were used to handle the fact that spike events are not differentiable [4, 5]. During the forward pass, neurons generate binary spikes based on the LIF dynamics. During backpropagation, the spike function is replaced with a smooth approximation so that gradients can propagate. Without a surrogate gradient, the spike function would have zero derivative almost everywhere, causing gradients to vanish and preventing effective weight updates. In that case, learning would stall and classification accuracy would remain near chance.

This setup makes it possible to directly compare a dense feed-forward network with a spike-based network under matched training conditions. By keeping the network architecture, parameter count, and dataset fixed, any differences in accuracy, sparsity, or computational cost can be attributed primarily to the use of continuous activations versus discrete spiking dynamics.

6 Results

6.1 Training loss comparison

Figure 3 compares the training loss of a standard feed-forward neural network (FFNN) and a spiking neural network (SNN) trained on the same MNIST task under matched conditions. The FFNN shows a rapid drop in loss early in training, while the SNN's loss decreases more gradually before converging later. This behavior reflects the added challenge of training networks based on discrete spikes [4].

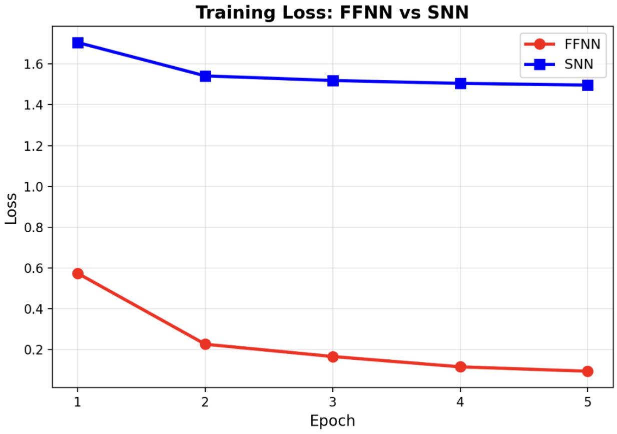

Figure 3: Training loss for the FFNN and SNN during MNIST training.

6.2 Training accuracy comparison

Training accuracy curves for the FFNN and SNN are shown in Figure 4. The FFNN achieves higher accuracy during the first training epoch. This demonstrates the advantage of dense, fully differentiable activations during early optimization. In contrast, the SNN shows slower initial learning, which is expected due to the non-differentiable nature of spike events and the use of surrogate gradients during backpropagation [4, 5].

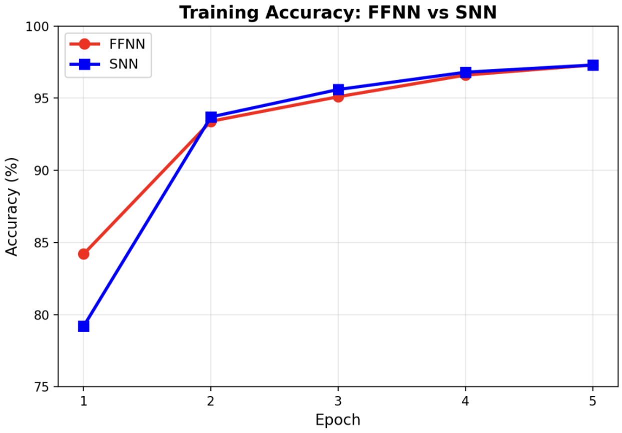

Figure 4: Training accuracy for FFNN and SNN on MNIST.

As training continues, the gap between the two models narrows. By later epochs, the SNN reaches accuracy close to that of the FFNN. This shows that spike-based networks can learn useful representations when trained with surrogate gradients. Although the SNN introduces additional optimization challenges early on, this does not prevent it from reaching comparable final performance on MNIST.

6.3 Classification accuracy and computational cost

Figures 5–7 present the final classification accuracy, estimated computational cost, and neuron activity sparsity for the feed-forward neural network (FFNN) and the spiking neural network (SNN).

Final classification accuracy for both models is shown in Figure 5. The FFNN achieves slightly higher accuracy than the SNN, but the difference is small. This result shows that the SNN can perform competitively on MNIST despite relying on sparse, event-driven computation, consistent with previous work on surrogate-gradient training of spiking networks [4, 5].

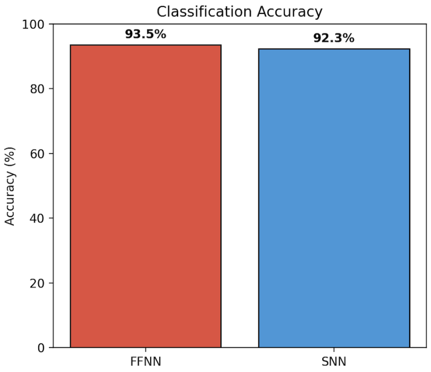

Figure 5: Final MNIST classification accuracy.

Figure 6 compares the computational cost of the two models. While the FFNN requires dense multiply–accumulate operations for every neuron and every input, the SNN requires substantially fewer operations. Quantitatively, the feed-forward network requires approximately 537,600 floating-point operations (FLOPs) per input sample, based on dense matrix multiply–accumulate operations across all layers. In contrast, the spiking network requires approximately 7,111 synaptic operations (SOPs) per sample. This corresponds to a reduction of roughly 76× in the number of operations.

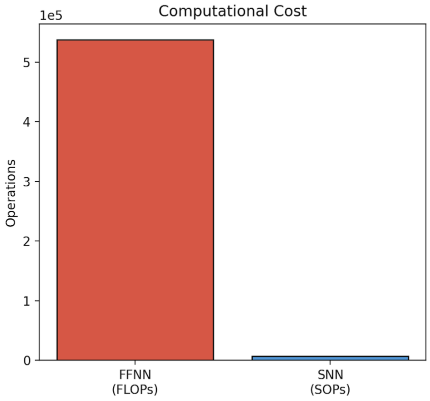

Figure 6: Estimated computational cost for the FFNN and SNN.

Neuron activity sparsity is shown in Figure 7. The spiking network exhibits a high proportion of silent neurons at each time step, whereas the feed-forward network activates a large fraction of its neurons for every input. Because inactive neurons in the SNN produce no spikes, a large fraction of potential synaptic updates never occur. Quantitatively, approximately 88.9% of neurons in the spiking network remain silent at each time step, so the corresponding synaptic updates are never executed. The resulting efficiency gain follows directly from this spike sparsity.

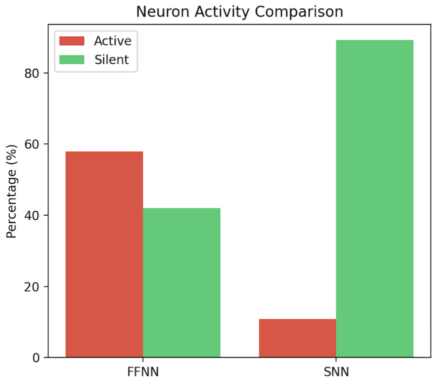

Figure 7: Neuron activity sparsity comparison

Figures 5–7 together illustrate the tradeoff between accuracy and computation in dense and spiking networks. Despite a small reduction in final accuracy, the SNN achieves performance close to that of the FFNN while requiring far fewer operations. This efficiency gain arises from spike sparsity: most neurons in the spiking network remain inactive at each time step, so many potential synaptic updates never occur. In contrast, the feed-forward network performs dense computations for all neurons regardless of their contribution. These results show that comparable classification performance can be achieved with significantly less computation when spike-driven activity is present, highlighting the potential of spiking networks for energy-efficient inference on event-driven hardware [1, 2].

7 Discussion

This work connects ideas from neuroscience, machine learning, and hardware-aware computing. As AI systems continue to grow in size and energy demand, approaches that rethink how computation is performed are becoming essential. Spiking neural networks offer a different way of thinking about neural computation, one that moves away from dense, always-active processing and toward sparse, event-driven computation inspired by biology. Showing that spiking networks can reach similar accuracy to standard feed-forward models while using far less activity suggests that this approach has real potential. These results support the idea that biologically inspired computation could play an important role in building more efficient and sustainable AI systems in the future [1, 4, 5].

8 Future Directions for Spiking Neural Networks

Spiking neural networks offer a potential path toward more energy-efficient computation, but their advantages depend on how well neuron models, training methods, and hardware work together. As shown in this work, spike sparsity alone is not sufficient; it must be supported by systems that can take advantage of event-driven computation [1, 2].

8.1 Neuromorphic hardware platforms

Neuromorphic hardware platforms such as Intel's Loihi and IBM's TrueNorth are designed specifically for spike-based computation, allowing neurons to communicate asynchronously and only perform computation when spikes occur [1, 2]. These architectures can reduce unnecessary operations compared to conventional CPUs and GPUs, but efficiency gains depend strongly on how well network structure and spike activity align with the hardware [1].

8.2 Hybrid and scalable SNN approaches

Hybrid ANN–SNN architectures offer a compromise that combines dense layers with spiking components for temporal processing or energy-efficient inference [2]. At the same time, scaling SNNs to larger datasets remains challenging due to longer simulation times and training instability on conventional hardware. Continued progress in surrogate-gradient methods and hardware-aware training will be necessary to extend SNNs beyond small benchmarks such as MNIST [4, 1].

Conclusions

This work shows that the leaky integrate-and-fire neuron provides a clear and interpretable link between biological membrane dynamics and artificial activation functions. The LIF firing-rate behavior highlights key differences between biological excitability and common ANN nonlinearities, particularly in how thresholding and saturation arise [3, 5]. At the network level, spiking neural networks built from LIF neurons achieve accuracy comparable to feed-forward baselines on MNIST while producing sparse spike activity, supporting the motivation for event-driven computation [1, 4]. Future work should focus on evaluating these models under hardware-realistic conditions and exploring architectures that maximize sparsity without sacrificing performance [1, 2].

References

[1] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel. Are SNNs really more energy-efficient than ANNs? An in-depth hardware-aware study. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(3):731–741, 2023. https://doi.org/10.1109/TETCI.2022.3214509.

[2] A. Jr. Spiking neural networks: The future of brain-inspired computing. *International Journal of Engineering Trends and Technology*, 73(10), 2025. https://doi.org/10.14445/22315381/ijett-v73i10p104.

[3] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition.* Cambridge University Press, 2014. https://neuronaldynamics.epfl.ch/online/Ch1.S3.html.

[4] M. Pfeiffer and T. Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in Neuroscience*, 12:774, 2018. https://pmc.ncbi.nlm.nih.gov/articles/PMC6209684/.

[5] S. Lu and A. Xu. Linear leaky-integrate-and-fire neuron model based spiking neural networks and its mapping relationship to deep neural networks. *Frontiers in Neuroscience*, 16:820158, 2022. https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2022.857513/full