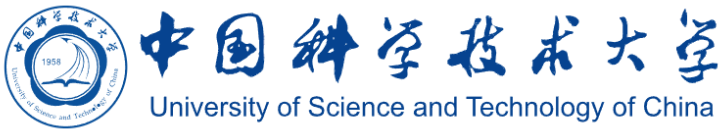
TQ-TTFS: High-Accuracy and Energy-Efficient Spiking Neural Networks Using Temporal Quantization Time-to-First-Spike Neuron

Yuxuan Yang, Zihao Xuan, and Yi Kang

University of Science and Technology of China, Hefei, China



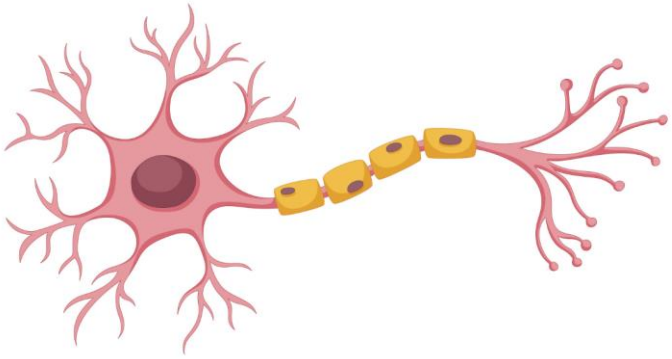
Outline

- Introduction and motivation
- Preliminaries
- TQ-TTFS neuron model
- Experiments and analysis
- Conclusion

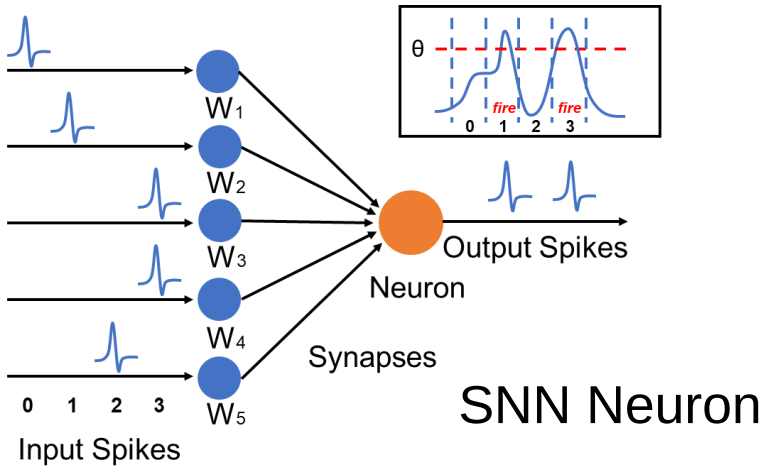
Outline

- Introduction and motivation
- Preliminaries
- TQ-TTFS neuron model
- Experiments and analysis
- Conclusion

Introduction and motivation



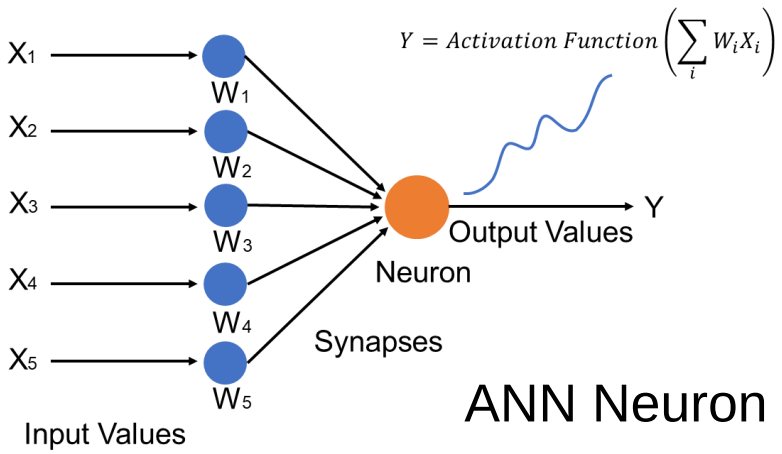
Neuron structure



SNN Neuron

SNN: discrete, sparse spikes

ANN: continuous, dense values



ANN Neuron

Introduction and motivation

	Rate coding	vs	TTFS coding
Spike numbers	Large		Few
Latency	High		Low
Energy efficiency	Low		High
Classification accuracy	High		Low

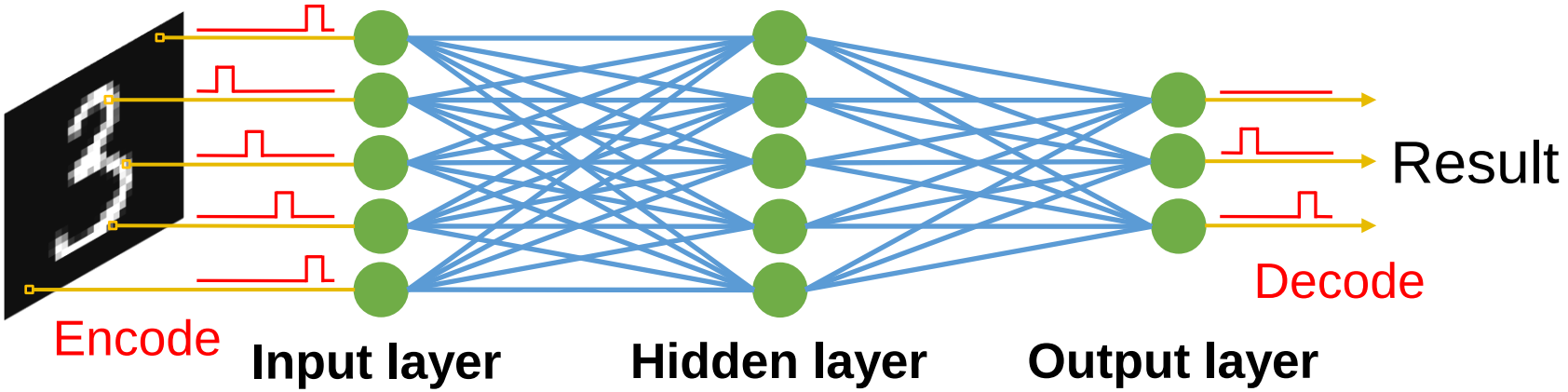
TTFS (Time-to-First-Spike) SNN + Quantification?

- Higher classification accuracy
- Higher energy efficiency

Outline

- Introduction and motivation
- Preliminaries
- TQ-TTFS neuron model
- Experiments and analysis
- Conclusion

Preliminaries



Encode: intensity-to-latency conversion

$$t = Round\left((T-1) * \left(1 - \frac{x}{255}\right)\right)$$

t: input time, T: total time steps, x: pixel intensity.

The higher the intensity, the earlier the input time.

Decode:

- The earliest firing neuron corresponds to the result
- If multiple neurons fire at the earliest or no neurons fire until the end, the neuron with the highest membrane potential corresponds to the result

Preliminaries

IF (Integrate-and-Fire) Neuron model

$$V_j^l[t] = \begin{cases} \sum_i W_{ji}^l S_j^{l-1}[t] + V_j^l[t-1], S_j^l[t-1] = 0 \\ \sum_i W_{ji}^l S_j^{l-1}[t] + V_{reset}, \qquad S_j^l[t-1] = 1 \end{cases}$$

$$S_j^l[t] = \begin{cases} 1, V_j^l[t] > \theta \\ 0, otherwise \end{cases}$$

W_{ji}^l: synapse between the i-th neuron in layer $l-1$ and the j-th neuron in layer l

S_j^l: spike of the j-th neuron in layer l

V_{reset}: resting potential

θ: threshold voltage

Outline

- Introduction and motivation
- Preliminaries
- TQ-TTFS neuron model
- Experiments and analysis
- Conclusion

TQ-TTFS neuron model

IF Neuron model + TTFS coding

$$V_j^l[t] = \begin{cases} \left(1 - F_j^l\right) * \sum_i W_{ji}^l S_j^{l-1}[t] + V_j^l[t-1], S_j^l[t-1] = 0 \\ \left(1 - F_j^l\right) * \sum_i W_{ji}^l S_j^{l-1}[t] + V_{reset}, \qquad S_j^l[t-1] = 1 \end{cases}$$

1

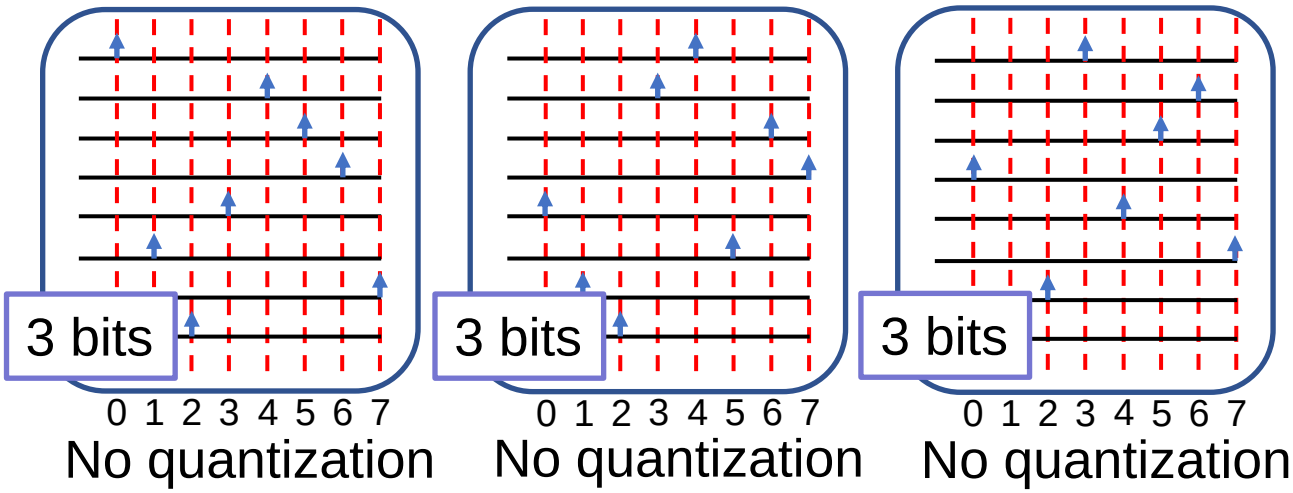
$$S_j^l[t] = \begin{cases} 1, \ V_j^l[t] > \theta \\ 0, otherwise \end{cases}, \ F_j^l = S_j^l[t] + F_j^l$$

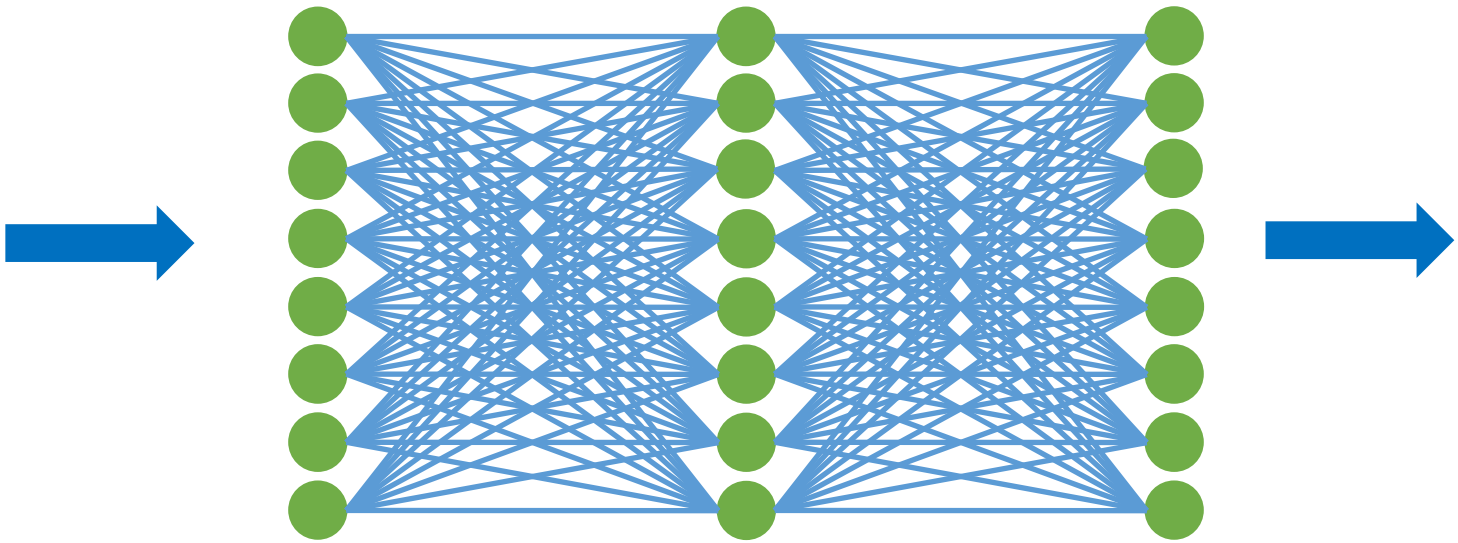
F: historical firing status of neurons

The initial value of F is 0, the neuron charges normally.

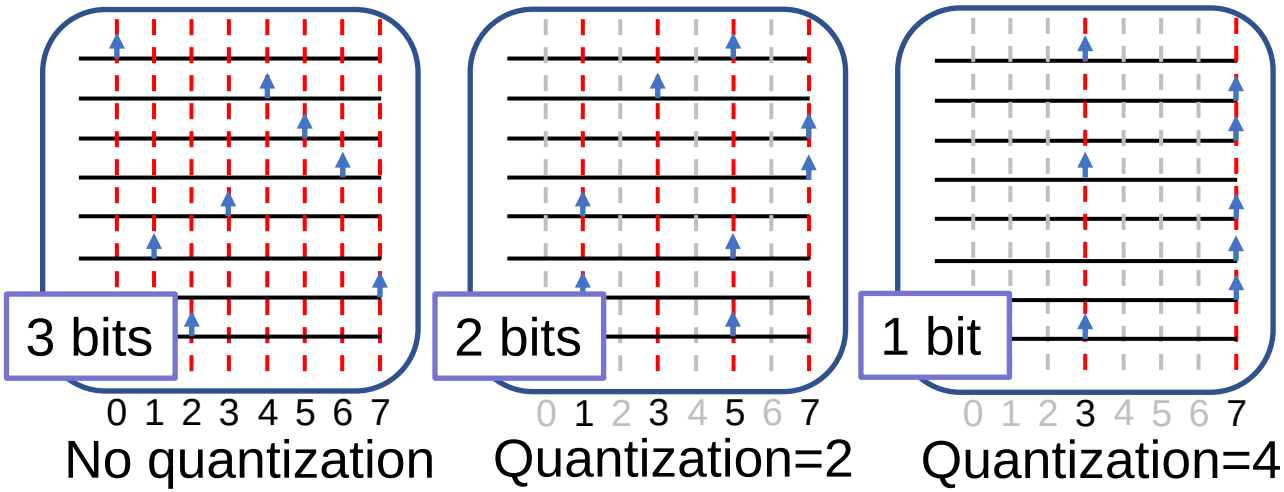
After neuron firing, F becomes 1, the neuron no longer charges.

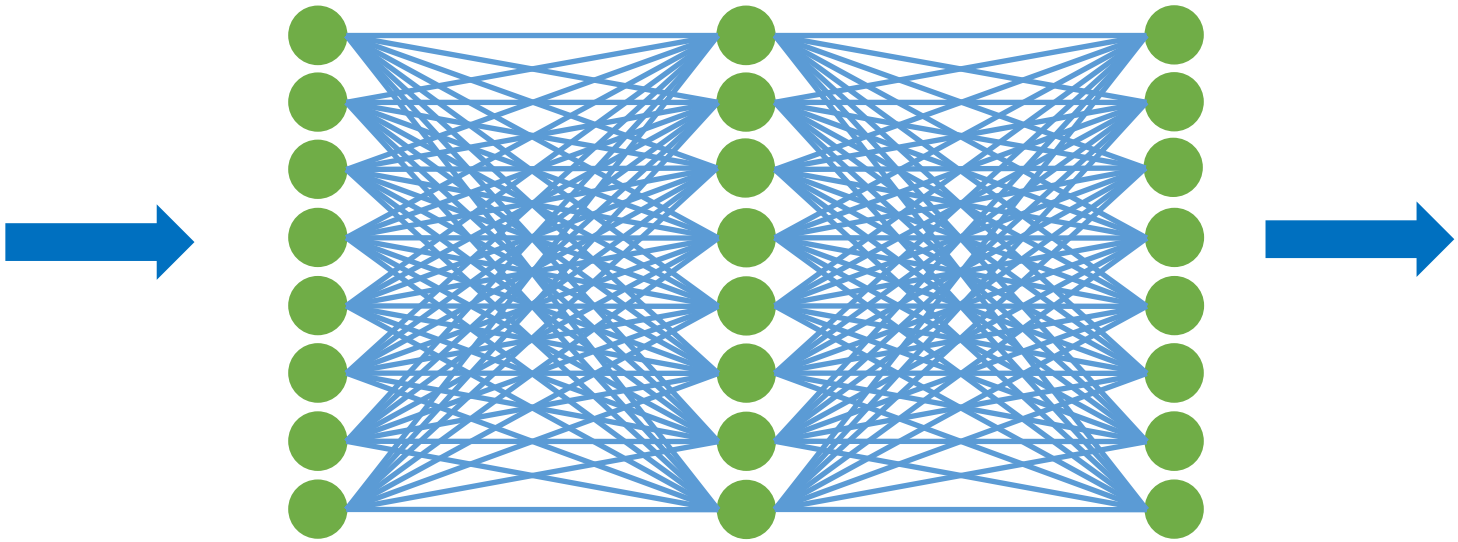
TQ-TTFS neuron model





TQ-TTFS neuron model





TQ-TTFS neuron model

IF Neuron model + TTFS coding + Temporal Quantification

$$V_j^l[t] = \begin{cases} \left(1 - F_j^l\right) * \sum_i W_{ji}^l S_j^{l-1}[t] + V_j^l[t-1], S_j^l[t-1] = 0 \\ V_{reset}, \qquad\qquad S_j^l[t-1] = 1 \end{cases}$$

$$S_j^l[t] = \begin{cases} 1, V_j^l[t] > \theta \ and \ (t+1)\%t_{quantization} = 0 \\ 0, otherwise \end{cases}, \quad F_j^l = S_j^l[t] + F_j^l$$

$t_{quantization}$: temporal quantization degree

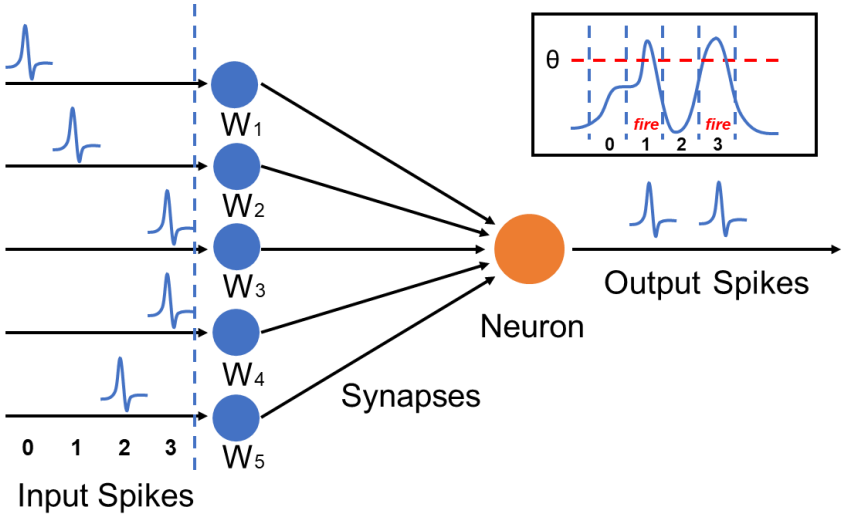
Continuous time steps ⟶ Evenly spaced time steps

Equivalent to a reduction in time steps

TQ-TTFS neuron model

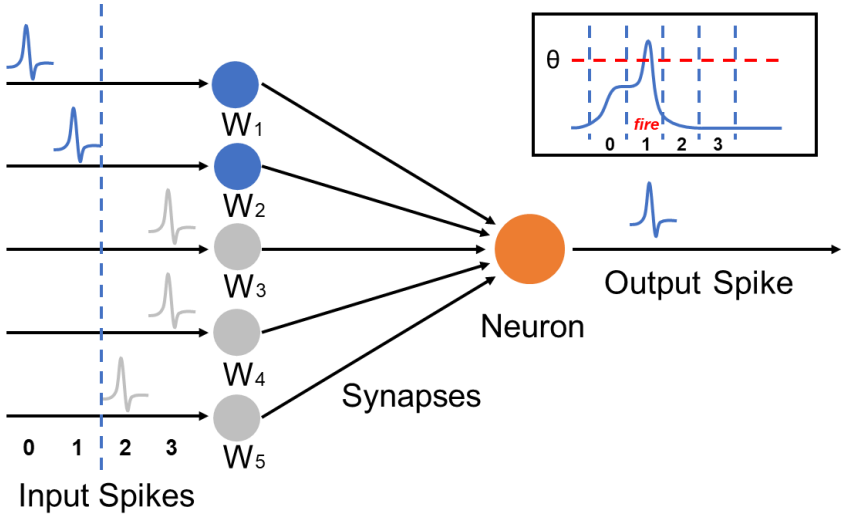
IF Neuron model

- Receive all spikes
- Integrate and Fire, no Leaky
- Fire multiple times



+ TTFS coding

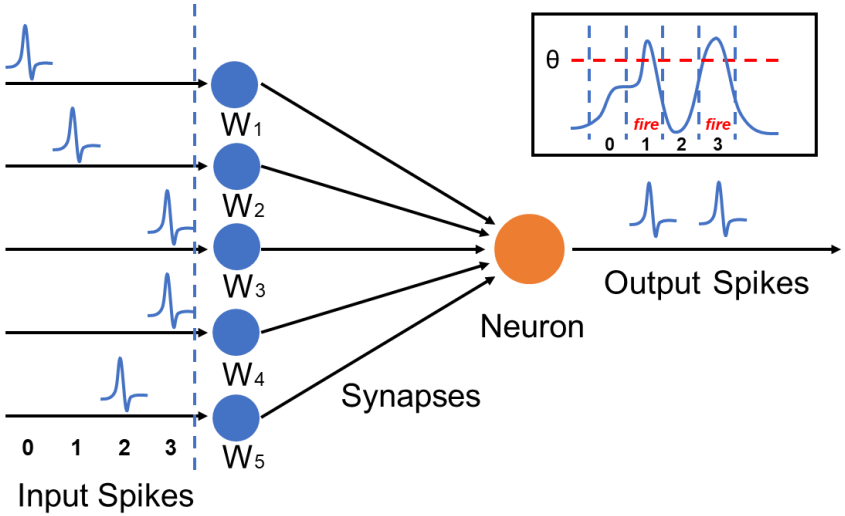
- Receive spikes before firing
- Integrate and Fire, no Leaky
- Fire once at most



TQ-TTFS neuron model

IF Neuron model

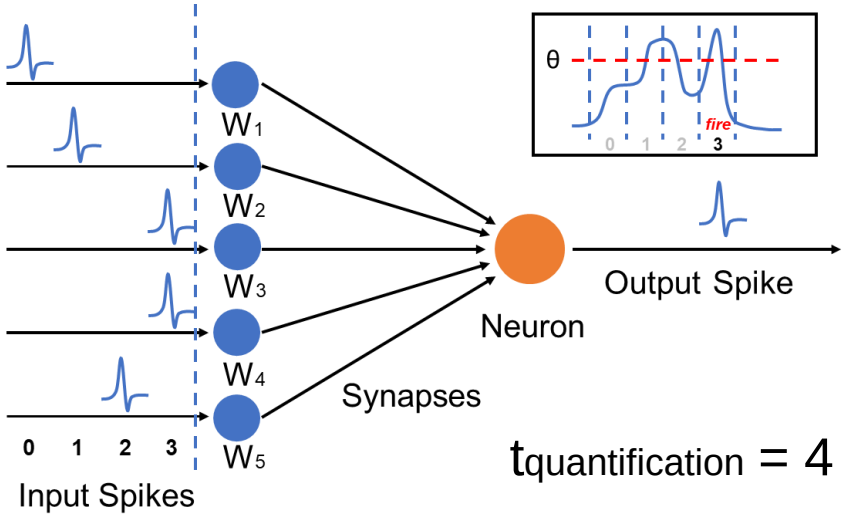
- Receive all spikes
- Integrate and Fire, no Leaky
- Fire multiple times



+ TTFS coding

+ Temporal Quantification

- Receive spikes before firing
- Integrate and Fire, no Leaky
- Fire once at most
- Can only fire when $(t+1)\%t_{quantification} = 0$



Outline

- Introduction and motivation
- Preliminaries
- TQ-TTFS neuron model
- Experiments and analysis
- Conclusion

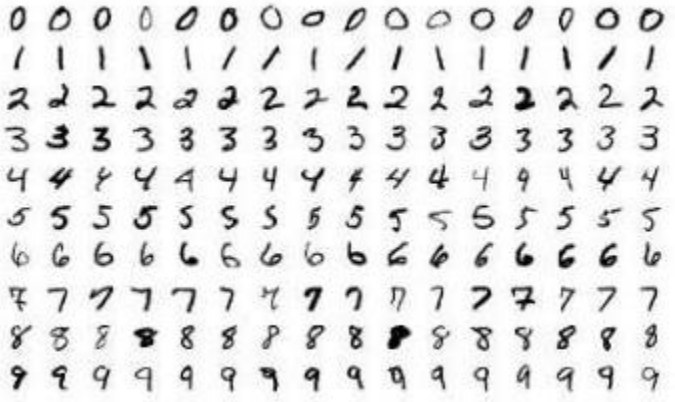
Experiments and analysis

MNIST

Content: handwritten digits from 0 to 9

Image size: 28*28, grayscale images

Dataset size: 60,000 images in training set,
10,000 images in test set



FashionMNIST

Content: fashion products from 10 categories

Image size: 28*28, grayscale images

Dataset size: 60,000 images in training set,
10,000 images in test set

Training method: surrogate gradient based backpropagation

Surrogate gradient function: Atan function

Experiments and analysis

Experiments on MNIST Dataset

Notation of a temporal quantized SNNs:

Table: input layer - layer1(Q_1) - ... - layern(Q_n) - output layer(Q_{n+1})

Figure: Q_1 - ... - Q_n - Q_{n+1} Q: quantization degree

SMLP: 784-400-10, total time steps = 8

- TQ-TTFS SMLP achieves a 0.5% accuracy improvement compared to the baseline SMLP
- TQ-TTFS SMLP achieves the best performance within the same neural network structure compared to other works, reaching 98.6%

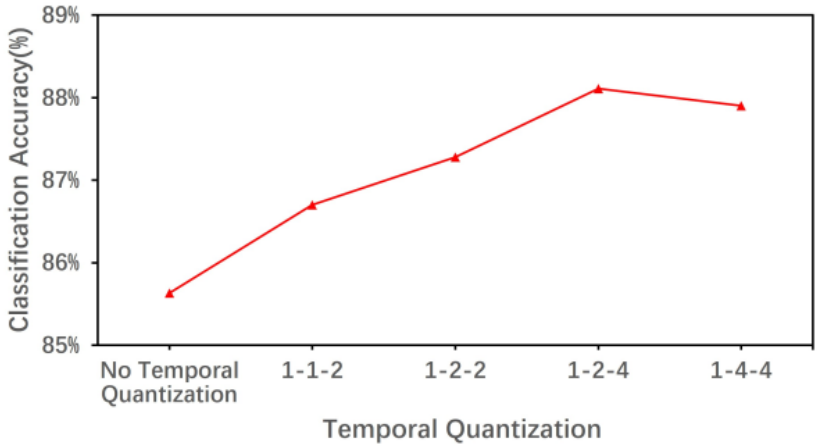
CLASSIFICATION ACCURACIES ON MNIST

Model	Network Architecture	Coding	ACC.
Mostafa [13]	784-800-10	Temporal	97.5%
Tavanaei et al.[14]	784-1000-10	Rate	96.6%
Comsa et al. [15]	784-340-10	Temporal	97.9%
S4NN [16]	784-600-10	Temporal	97.4%
BS4NN [5]	784-600-10	Temporal	97.0%
STDBP [6]	784-400-10	Temporal	98.1%
STDBP [6]	784-1000-10	Temporal	98.5%
Baseline SMLP	784-400-10 w/o Temporal Quantization	Temporal	98.1%
TQ-TTFS/SMLP	784-400(1)-10(2) w Temporal Quantization	Temporal	**98.6%**

Experiments and analysis

Experiments on FashionMNIST Dataset

SMLP: 784-400-400-10, total time steps = 16



- Best TQ-TTFS SMLP performance at 1-2-4, reaching 88.1%
- Best TQ-TTFS SMLP achieves 2.5% accuracy improvement compared to the baseline SMLP

CLASSIFICATION ACCURACIES ON FASHIONMNIST

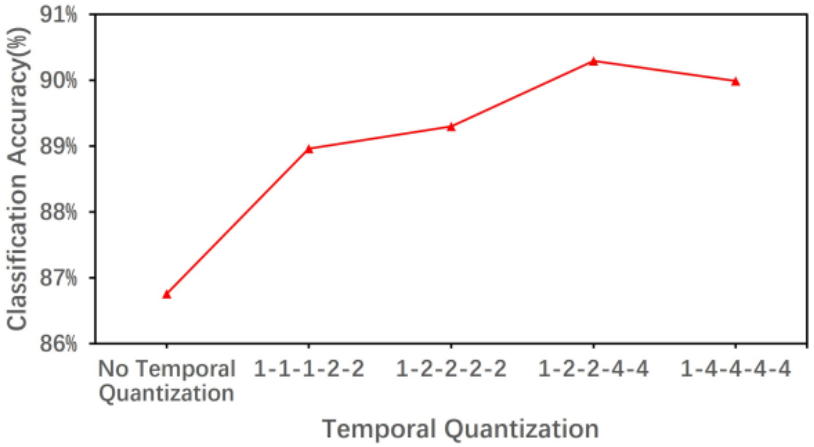
Model	Network Architecture	Coding	ACC.
S4NN [16]	784-1000-10	Temporal	88.0%
BS4NN [5]	784-1000-10	Temporal	87.3%
STDBP [6]	28×28-16C5[a]-P2[b]-32C5-P2-800-128-10	Temporal	90.1%
STDBP [6]	784-1000-10	Temporal	88.1%
Hao et al. [17]	784-6000-10	Rate	85.3%
Ranjan et al.[18]	784-400-400-10	Rate	89.0%
Baseline SMLP	784-400-400-10 w/o Temporal Quantization	Temporal	85.6%
TQ-TTFS/SMLP	784-400(1)-400(2)-10(4) w Temporal Quantization	Temporal	88.1%
Baseline SCNN	28×28-16C5-P2-32C5-P2-800-128-10 w/o Temporal Quantization	Temporal	86.7%
TQ-TTFS/SCNN	28×28-16C5(1)-P2-32C5(2)-P2-800(2)-128(4)-10(4) w Temporal Quantization	Temporal	**90.2%**

[a]16C5: convolution layer with 16 of the 5×5 filters.
[b]P2: pooling layer with 2×2 filters.

Experiments and analysis

Experiments on FashionMNIST Dataset

SCNN: 28×28-16C5-P2-32C5-P2-800-128-10, total time steps = 16



- Best TQ-TTFS SCNN performance at 1-2-2-4-4, reaching 90.2%
- Best TQ-TTFS SCNN achieves 3.5% accuracy improvement compared to the baseline SCNN
- The highest classification accuracies among the known TTFS coding SNN

CLASSIFICATION ACCURACIES ON FASHIONMNIST

Model	Network Architecture	Coding	ACC.
S4NN [16]	784-1000-10	Temporal	88.0%
BS4NN [5]	784-1000-10	Temporal	87.3%
STDBP [6]	28×28-16C5[a]-P2[b]-32C5-P2-800-128-10	Temporal	90.1%
STDBP [6]	784-1000-10	Temporal	88.1%
Hao et al. [17]	784-6000-10	Rate	85.3%
Ranjan et al.[18]	784-400-400-10	Rate	89.0%
Baseline SMLP	784-400-400-10 w/o Temporal Quantization	Temporal	85.6%
TQ-TTFS/SMLP	784-400(1)-400(2)-10(4) w Temporal Quantization	Temporal	88.1%
Baseline SCNN	28×28-16C5-P2-32C5-P2-800-128-10 w/o Temporal Quantization	Temporal	86.7%
TQ-TTFS/SCNN	28×28-16C5(1)-P2-32C5(2)-P2-800(2)-128(4)-10(4) w Temporal Quantization	Temporal	**90.2%**

[a] 16C5: convolution layer with 16 of the 5×5 filters.
[b] P2: pooling layer with 2×2 filters.

Experiments and analysis

Energy Efficiency Analysis

The energy consumption of inferring a single image

$$E = T * E_{AC} * \sum_{l} FLOPs(l) * R_{Method}(l)$$

$$R_{Method}(l) = \mathbb{E}\left[\frac{\#spikes\ of\ l\ th\ layer}{\#neurons\ of\ l\ th\ layer}\right]$$

$$FLOPs(l) = \begin{cases} {k^{(l)}}^2 * W^{(l)} * H^{(l)} * C_{in}^{(l)} * C_{out}^{(l)}, & Conv\ layer \\ C_{in}^{(l)} * C_{out}^{(l)}, & Linear\ layer \end{cases}$$

T: average time steps for the SNN to infer a single image

E_{AC}: energy consumption per additive operation, $E_{AC} = 0.9pj$ (45nm CMOS)

$R_{Method}(l)$: firing rate of the l-th layer

$FLOPs$: number of floating-point operations in l-th layer

Experiments and analysis

Energy Efficiency Analysis

Combining T and R_{Method}

$$E = E_{AC} * \sum_{l} FLOPs(l) * \frac{total\ spikes\ of\ l\ th\ layer}{neurons\ of\ l\ th\ layer}$$

For a regular TTFS SNN

inference time steps for a single image: T_0

frequency for all layer: f_0

Latency for inferring an image:

$$Latency = \frac{T_0}{f_0}$$

For a temporal quantized SNN

inference time steps for a single image: T_1

temporal quantization for l-th layer: Q_l

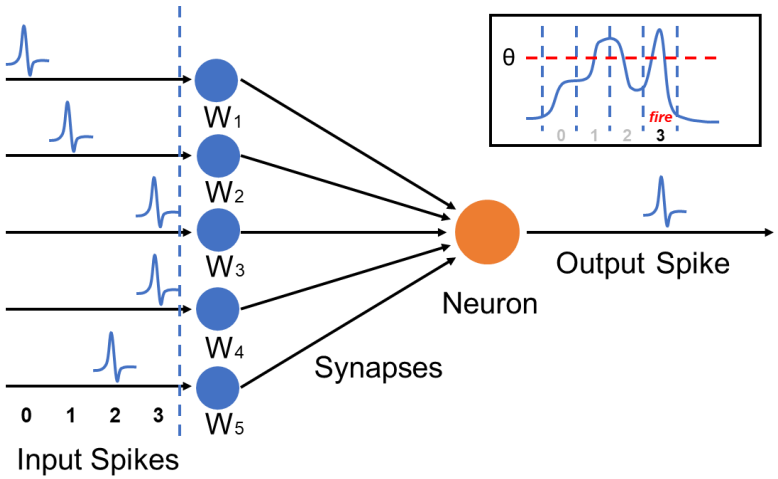
frequency for l-th layer: f_l

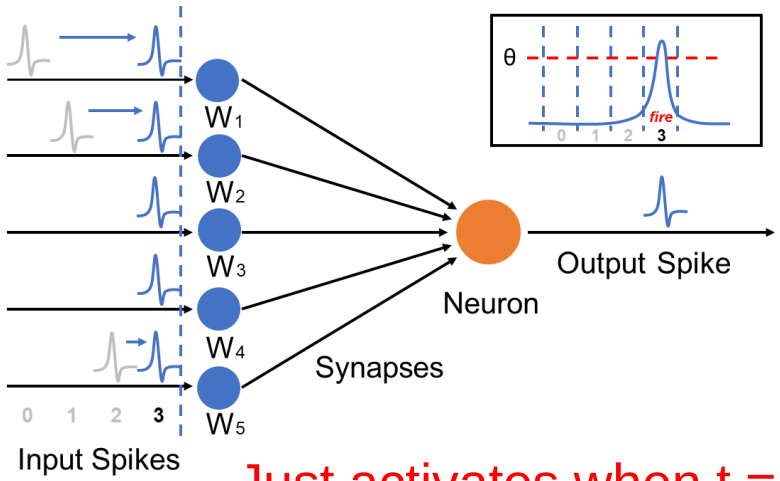
Latency for inferring an image:

$$Latency = \frac{\frac{T_1}{Q_l}}{f_l} = \frac{T_1}{Q_l f_l}$$

Experiments and analysis

Energy Efficiency Analysis





Just activates when t = 3.

Store spikes and receive them together.

For a temporal quantized SNN

inference time steps for a single image: T_1

temporal quantization for l-th layer: Q_l

frequency for l-th layer: f_l

Latency for inferring an image:

$$Latency = \frac{\frac{T_1}{Q_l}}{f_l} = \frac{T_1}{Q_l f_l}$$

Experiments and analysis

Energy Efficiency Analysis

If the latency is equal, i.e., the inference speed is equal

$$\frac{T_0}{f_0} = \frac{T_1}{Q_l f_l}$$

So

$$f_l = \frac{T_1 f_0}{T_0 Q_l}$$

Set α to be a frequency coefficient to describes the relationship between frequency f and energy consumption E, and $\alpha \propto f$, Then

$$\alpha_l = \frac{T_1 \alpha_0}{T_0 Q_l}$$

For the regular TTFS SNN, define frequency coefficient $\alpha_0 = 1$.

Experiments and analysis

Energy Efficiency Analysis

E can be classified as dynamic power consumption, so $E \propto f$, therefore

$$E \propto \alpha$$

Therefore, considering the influence of frequency, the formula of energy consumption E is revised to

$$E = E_{AC} * \sum_{l} FLOPs(l) * \alpha_l * \frac{total\ spikes\ of\ l\ th\ layer}{neurons\ of\ l\ th\ layer}$$

Where

$$\alpha_l = \frac{T_1 \alpha_0}{T_0 Q_l}$$

Evaluate the energy consumption of SCNN on FashionMNIST dataset with different temporal quantization configurations.

Experiments and analysis

Energy Efficiency Analysis

NEURONS AND FLOPS IN EACH LAYER

Data	Layer1	Layer2	Layer3	Layer4	Output Layer
Neurons	12544	6272	800	128	10
FLOPs	313600	2508800	1254400	102400	1280

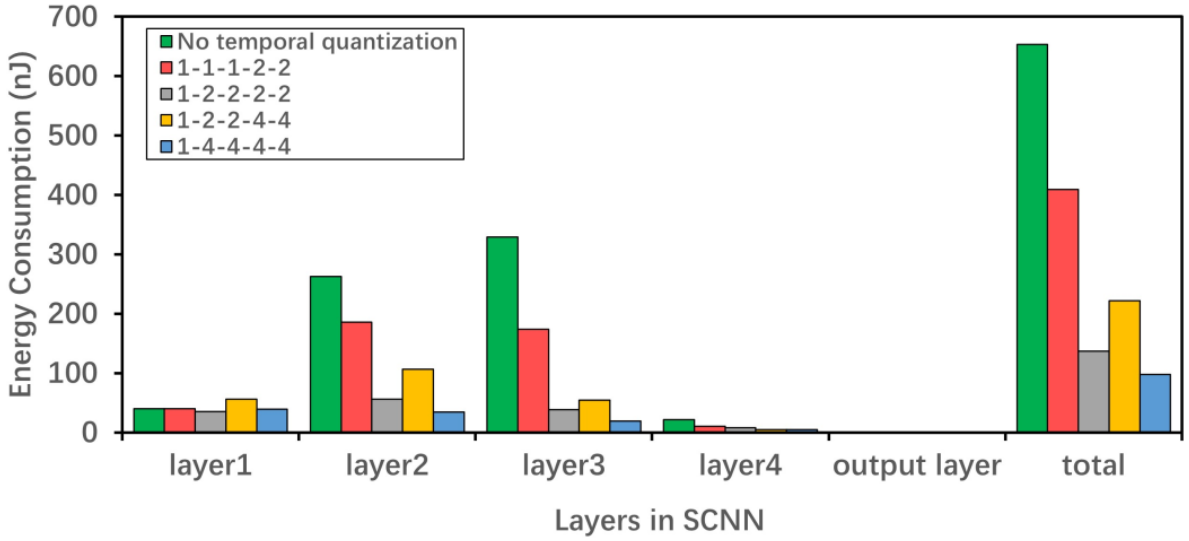
SCNN: 28×28-16C5-P2-32C5-P2-800-128-10

- Fewer spikes for inferring an image
- Temporal quantization increases the inference time steps
- Smaller α and lower frequencies in temporal quantization layer

Layers		Spikes	Inference time steps	α
Layer1	No temporal quantization	1763.94	4.766	1
	1-1-1-2-2	1538.94	5.471	1.147
	1-2-2-2-2	1318.03	5.609	1.176
	1-2-2-4-4	1663.24	7.102	1.490
	1-4-4-4-4	1217.15	6.880	1.443
Layer2	No temporal quantization	730.00	4.766	1
	1-1-1-2-2	448.94	5.471	1.147
	1-2-2-2-2	265.84	5.609	0.588
	1-2-2-4-4	397.94	7.102	0.745
	1-4-4-4-4	266.42	6.880	0.360
Layer3	No temporal quantization	233.35	4.766	1
	1-1-1-2-2	107.24	5.471	1.147
	1-2-2-2-2	45.89	5.609	0.588
	1-2-2-4-4	52.11	7.102	0.745
	1-4-4-4-4	37.13	6.880	0.360
Layer4	No temporal quantization	30.22	4.766	1
	1-1-1-2-2	24.94	5.471	0.573
	1-2-2-2-2	18.50	5.609	0.588
	1-2-2-4-4	17.29	7.102	0.372
	1-4-4-4-4	18.36	6.880	0.360
Output Layer	No temporal quantization	1.09	4.766	1
	1-1-1-2-2	1.05	5.471	0.573
	1-2-2-2-2	1.05	5.609	0.588
	1-2-2-4-4	1.02	7.102	0.372
	1-4-4-4-4	1.02	6.880	0.360

Experiments and analysis

Energy Efficiency Analysis



- Significant energy consumption reduction in time quantification layer
- Best energy efficiency at 1-4-4-4-4, reduces energy consumption by 85.0% compared to regular SCNN (Blue bar chart)
- At 1-2-2-4-4, achieve the highest classification accuracy of 90.2%, with a reduction of 66.0% energy consumption equivalent to 2.94× energy efficiency Blue bar chart (Yellow bar chart)

Outline

- Introduction and motivation
- Preliminaries
- TQ-TTFS neuron model
- Experiments and analysis
- Conclusion

Conclusion

- We propose an innovative way to perform temporal quantization for TTFS (TQ-TTFS) neuron model, which transform continuous time steps into evenly spaced time steps
- By applying temporal quantization, we significantly improve the classification accuracy of TTFS SNN
- TQ-TTFS SNN can operate at lower frequencies without increasing latency, resulting in better energy efficiency compared to regular SNN
- On the FashionMNIST dataset, we achieved a classification accuracy of 90.2% using a TQ-TTFS SCNN with 2.94× energy efficiency compared to the regular SNN

Thank You