Shahid Beheshti University
Faculty of Mathematical Sciences
Department of Data and Computer Science

Master thesis defence

Self-supervised learning on spiking neural networks

Supervisors : Dr. Hadi Farahani , Dr. Saeed Reza Kheradpisheh

By : Yeganeh Bahari Asl

September 2023

Road map

1

Spiking neural network

- introduction
- Spike encoding
- Neuron model
- Training SNN

2

Self-supervised learning

- Introduction
- Contrastive learning(simclr)
- Loss function

3

implementation

- Main idea
- Data augmentation
- Network architecture
- Evaluation metric

4

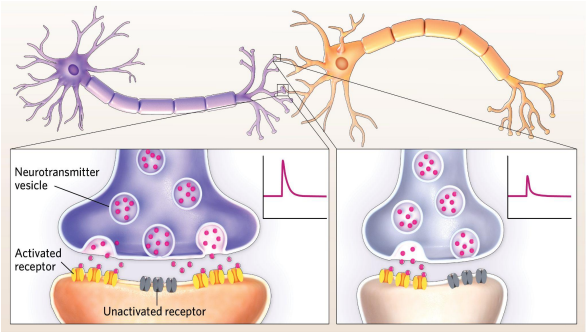
result

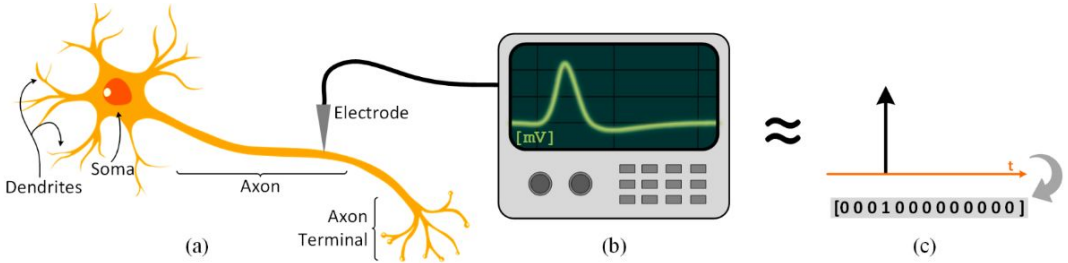
- Contrastive loss
- KNN
- classifier
- Research perspective

Introduction

Spikes: Biological neurons process and communicate via spikes through synaptic gates, which are **electrical impulses** of approximately **100 mV** in amplitude.

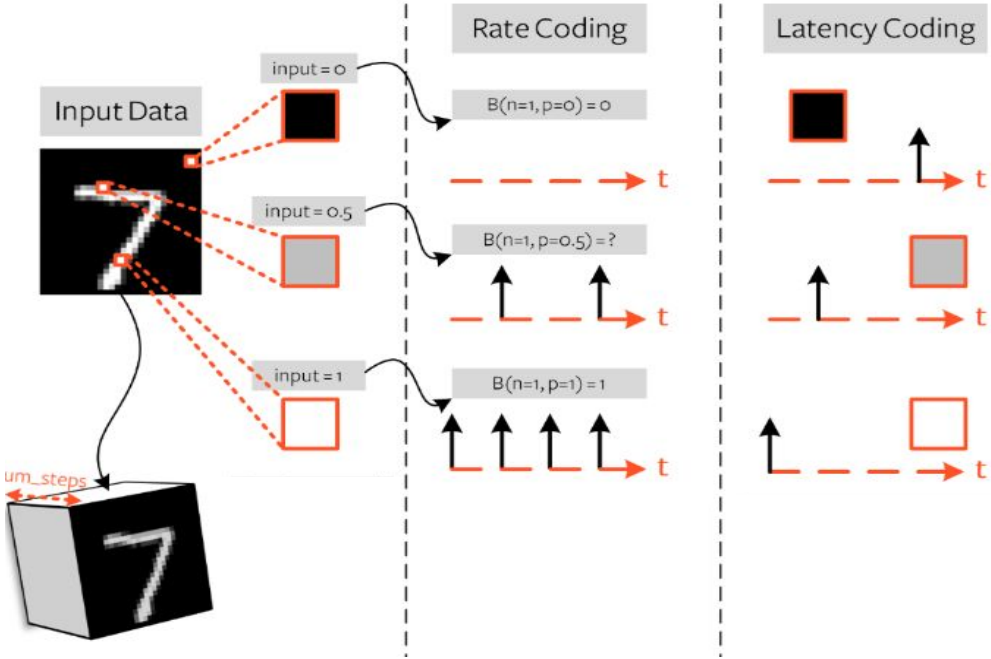
Sparsity: Neurons spend most of their time at rest, silencing most activations to *zero* at any given time.





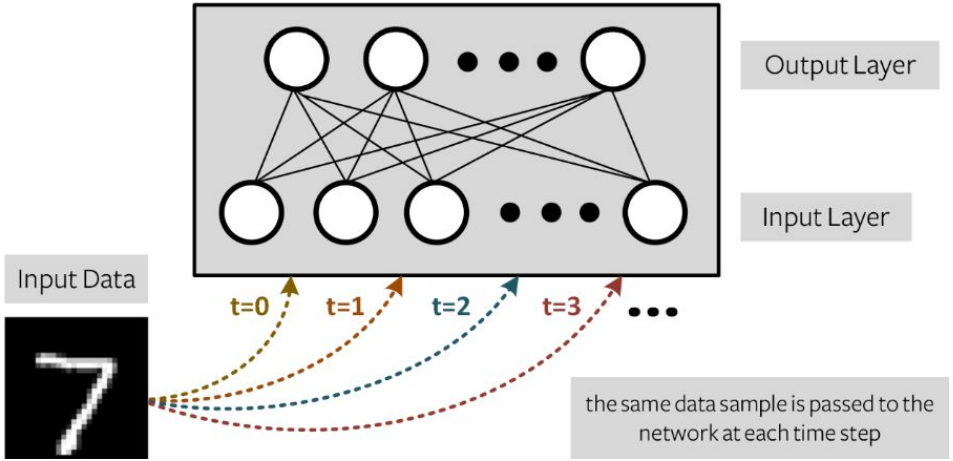
Spike encoding

1. ***Rate coding*** uses input features to determine spiking **frequency**
2. ***Latency coding*** uses input features to determine spike **timing**



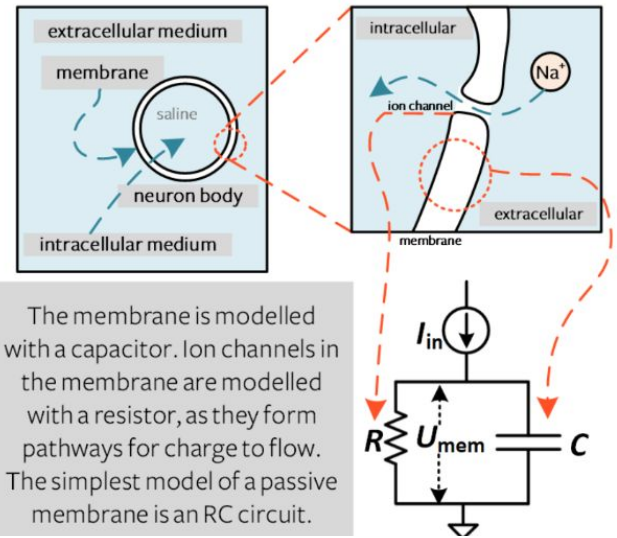
Spike encoding

Spiking Neural Networks (SNNs) are made to exploit time-varying data. And yet, MNIST and CIFAR10 are not a time-varying dataset. We need to repeatedly pass the same training sample to the network at each time step.

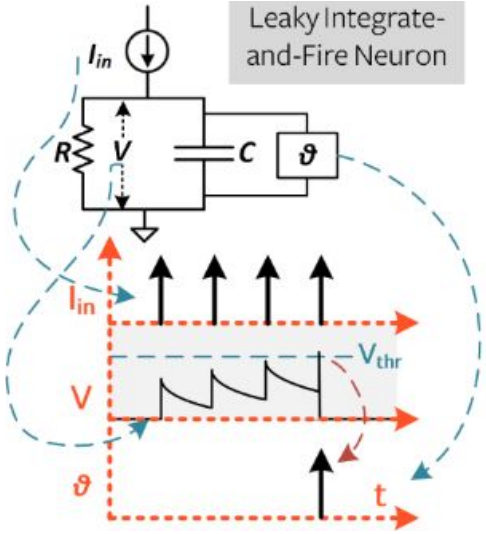


Neuron model (Leaky integrate and fire neuron)

- It takes the sum of weighted inputs, much like the artificial neuron.
- integrate the input over time with a leakage, much like an RC circuit.
- If the integrated value exceeds a threshold, then the LIF neuron will emit a voltage spike.

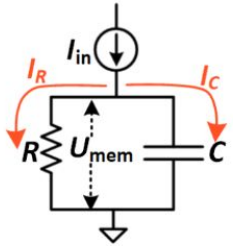


The membrane is modelled with a capacitor. Ion channels in the membrane are modelled with a resistor, as they form pathways for charge to flow. The simplest model of a passive membrane is an RC circuit.



When an input current is applied, the membrane voltage increases with time until it reaches a constant threhsold V_{th} at which point a delta function spike occurs, and the membrane is reset. ϑ is used as the output signal; membrane potential is an internal state.

Neuron model (Leaky integrate and fire neuron)



$$I_{in}(t) = I_R + I_C$$

$$\Rightarrow I_{in}(t) = \frac{U_{mem}(t)}{R} + C\frac{dU_{mem}(t)}{dt}$$

$$\Rightarrow I_{in}(t)R = U_{mem}(t) + RC\frac{dU_{mem}(t)}{dt}$$

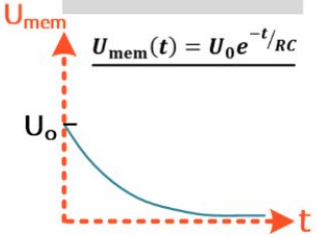
$$\Rightarrow U_{mem}(t) = I_{in}(t)R + c_1 e^{-t/RC}$$

where at t=0, $U_{mem}(t) = U_0$

$$\Rightarrow c_1 = U_0 - I_{in}(t)R$$

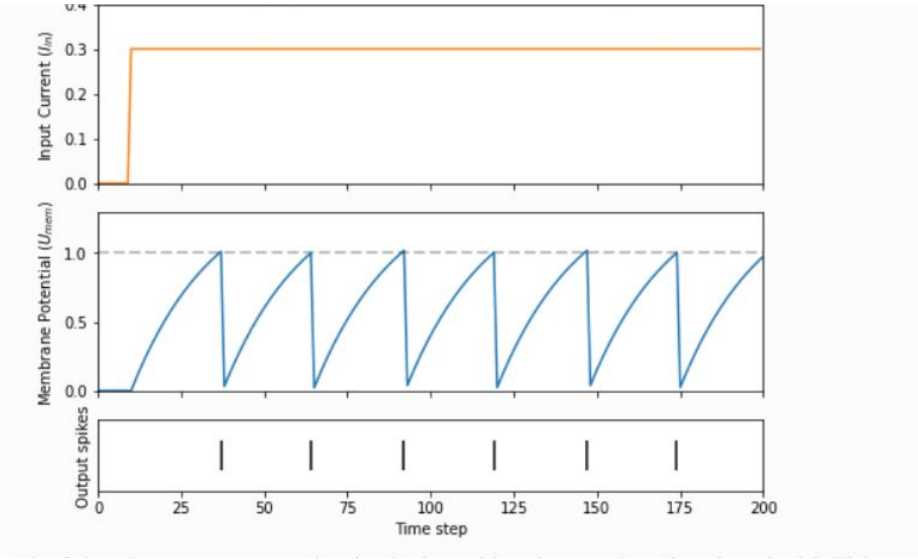
$$\Rightarrow U_{mem}(t) = I_{in}(t)R + [U_0 - I_{in}(t)R]e^{-t/RC}$$

If the input $I_{in}(t) = 0$:



$$U_{mem}(t) = U_0 e^{-t/RC}$$

In absence of input, the membrane potential will start at $U_{mem}=U_o$ and exponentially decay with a time constant τ=RC



Training SNN

THE NON-DIFFERENTIABILITY OF SPIKES

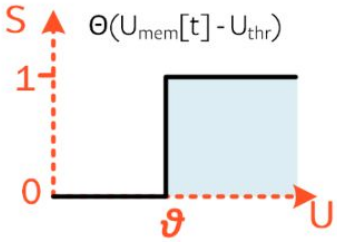
$$U(t+\Delta t) = (1-\frac{\Delta t}{\tau})U(t) + \frac{\Delta t}{\tau} I_{\text{in}}(t)R$$

$$\beta = (1-\frac{\Delta t}{\tau})$$

$$U[t+1] = \underbrace{\beta U[t]}_{\text{decay}} + \underbrace{WX[t+1]}_{\text{input}} - \underbrace{S[t]U_{\text{thr}}}_{\text{reset}}$$

$$S[t] = \begin{cases} 1, & \text{if } U[t] > U_{\text{thr}} \\ 0, & \text{otherwise} \end{cases}$$

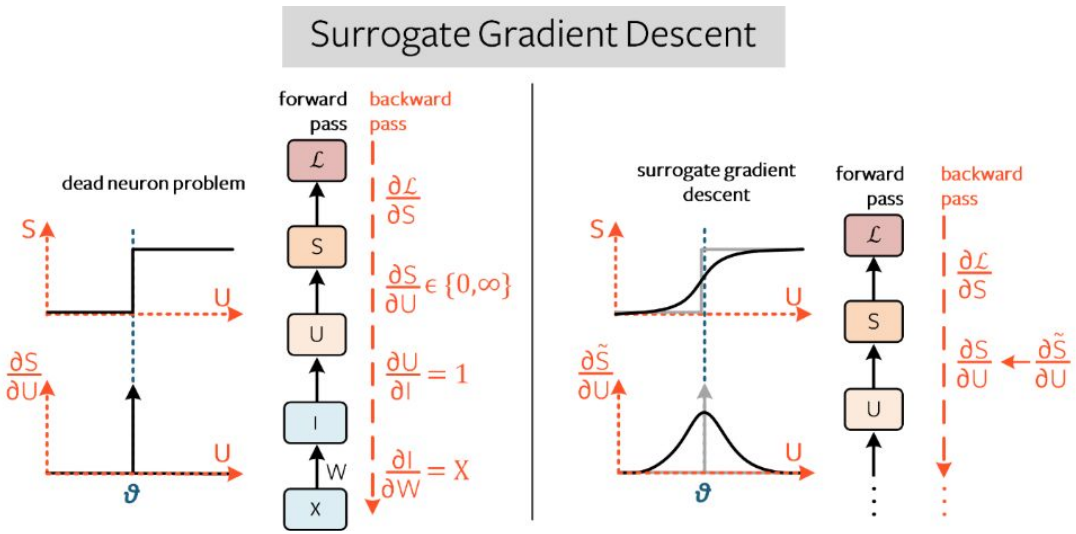
Mathematical Description of Spikes



If the membrane potential is greater than the threhsold $U_{thr} = \vartheta$, then $S_{out} = 1$.

Otherwise, $S_{out} = 0$.

Training SNN(Surrogate gradient descent)



$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial S} \underbrace{\frac{\partial S}{\partial U}}_{\{0,\infty\}} \frac{\partial U}{\partial I} \frac{\partial I}{\partial W}$$

$$S[t] = \Theta(U[t] - U_{\text{thr}})$$

$$\frac{\partial S}{\partial U} = \delta(U - U_{\text{thr}}) \in \{0, \infty\}$$

$$\tilde{S} = \frac{U_{OD}}{1 + k|U_{OD}|}$$

$$\frac{\partial \tilde{S}}{\partial U} = \frac{1}{(k|U_{OD}| + 1)^2}$$

$$\left.\frac{\partial \tilde{S}}{\partial U}\right|_{k \to \infty} = \delta(U - U_{\text{thr}})$$

Road map

1

Spiking neural network

- introduction
- Spike encoding
- Neuron model
- Training SNN

2

Self-supervised learning

- Introduction
- Contrastive learning(simclr)
- Loss function

3

implementation

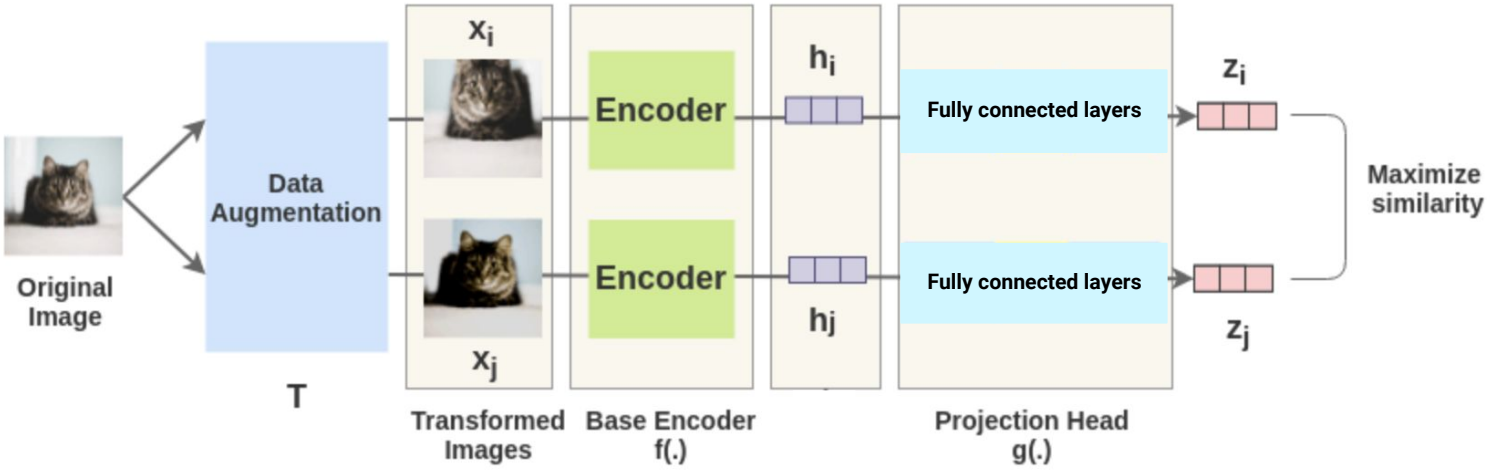
- Main idea
- Data augmentation
- Network architecture
- Evaluation metric

4

result

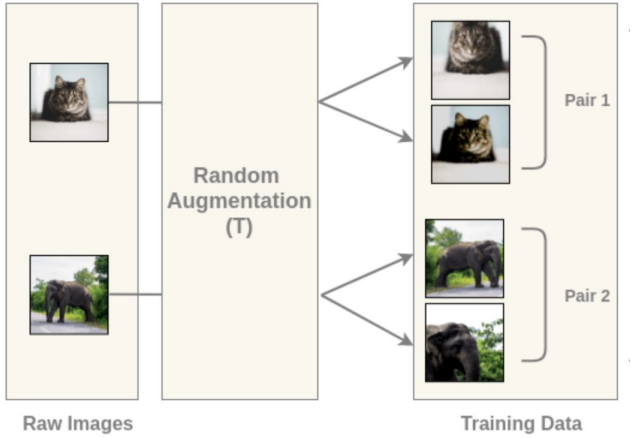
- Contrastive loss
- KNN
- classifier
- Research perspective

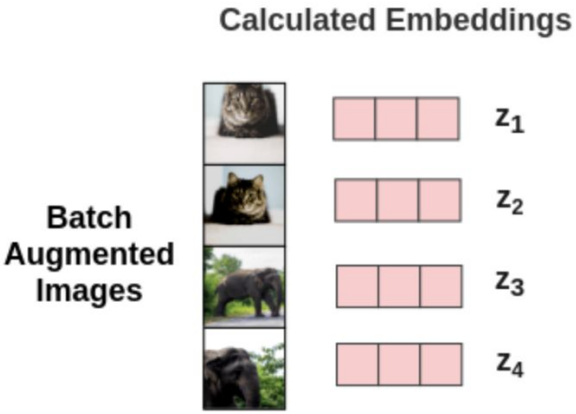
Introduction



Contrastive learning (simclr)

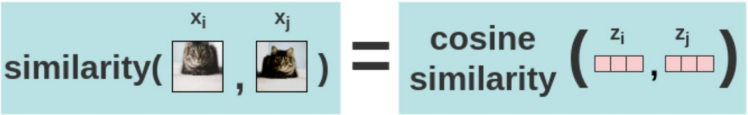
- Simclr : simple framework of contrastive learning
- Use unlabeled data
- Augment data
- Generate representation for each data





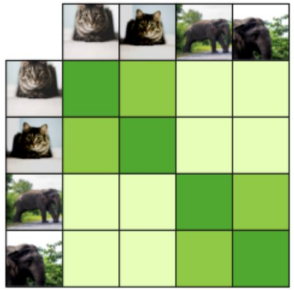
Loss function

Similarity Calculation of Augmented Images



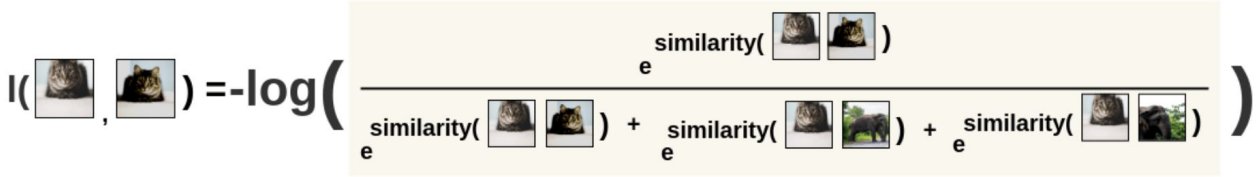
$$s_{i,j} = \frac{z_i^T z_j}{(\tau ||z_i|| ||z_j||)}$$

Pairwise cosine similarity

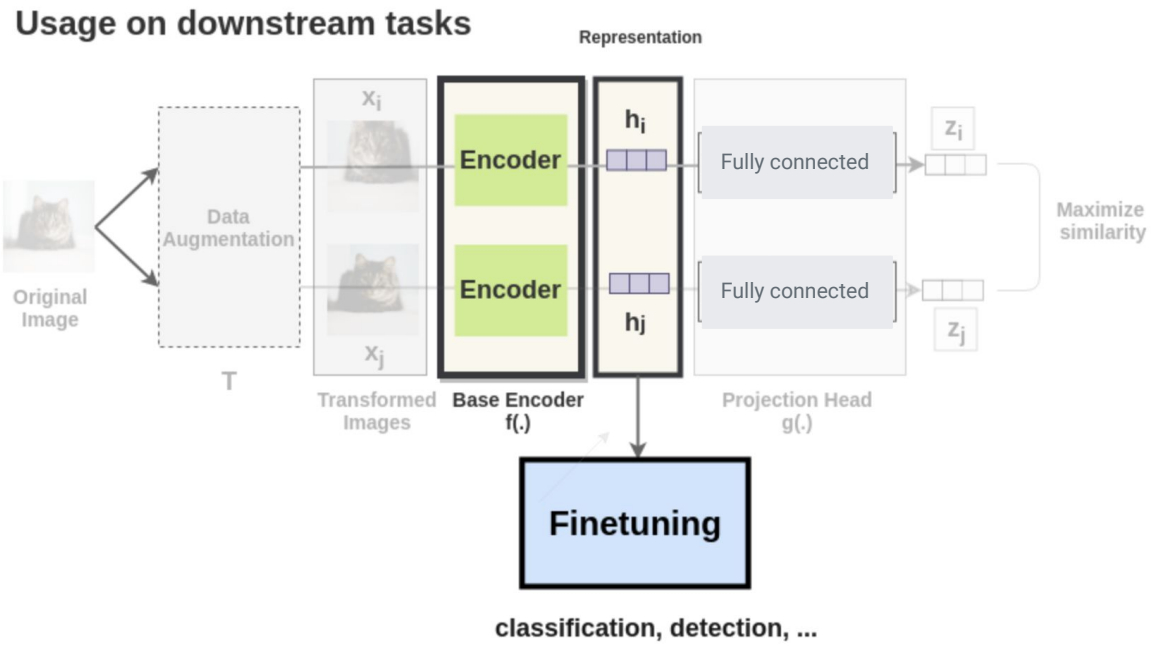


Normalized Temperature-Scaled Cross-Entropy Loss

$$l(i,j) = -log \frac{exp(s_{i,j})}{\sum_{k=1}^{2N} l_{[k!=i]} exp(s_{i,k})}$$



Contrastive learning (simclr)



Road map

1

Spiking neural network

- introduction
- Spike encoding
- Neuron model
- Training SNN

2

Self-supervised learning

- Introduction
- Contrastive learning(simclr)
- Loss function

3

implementation

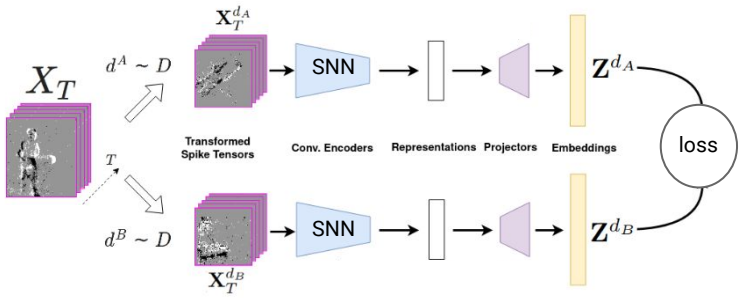
- Main idea
- Data augmentation
- Network architecture
- Evaluation metric

4

result

- Contrastive loss
- KNN
- classifier
- Research perspective

What is the main idea?

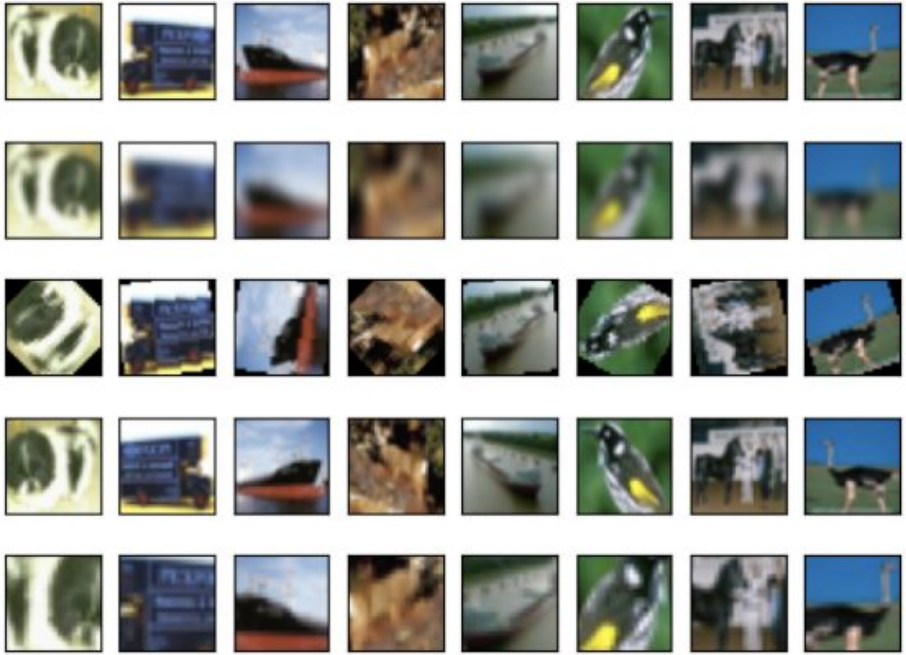


self-supervised + **Spiking neural network**

- **Use unlabeled data** (less need to pay money and take time for labeling)
- **Use less memory** (sparsity of spikes tensor)
- **Use less energy** (neurons are mostly at rest)

Data Augmentation

- Gaussian blur
- Rotate
- Flip
- crop



Network architecture

LIF

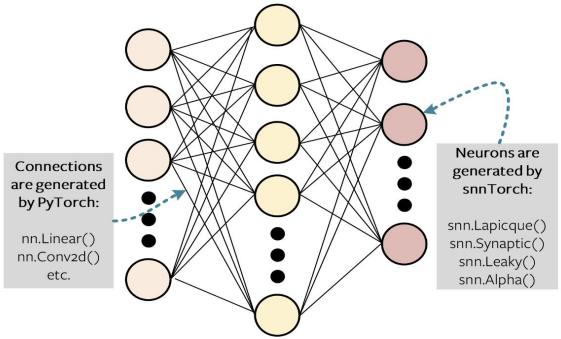
Use snntorch package for modeling lif neuron model with surrogate gradient descent.
Each neuron has a membrane potential and spikes as outputs

Convolutional

Use pytorch to model the connection between lif neuron with convolutional structure in order to extract features of augmented data

Fully connected classifier

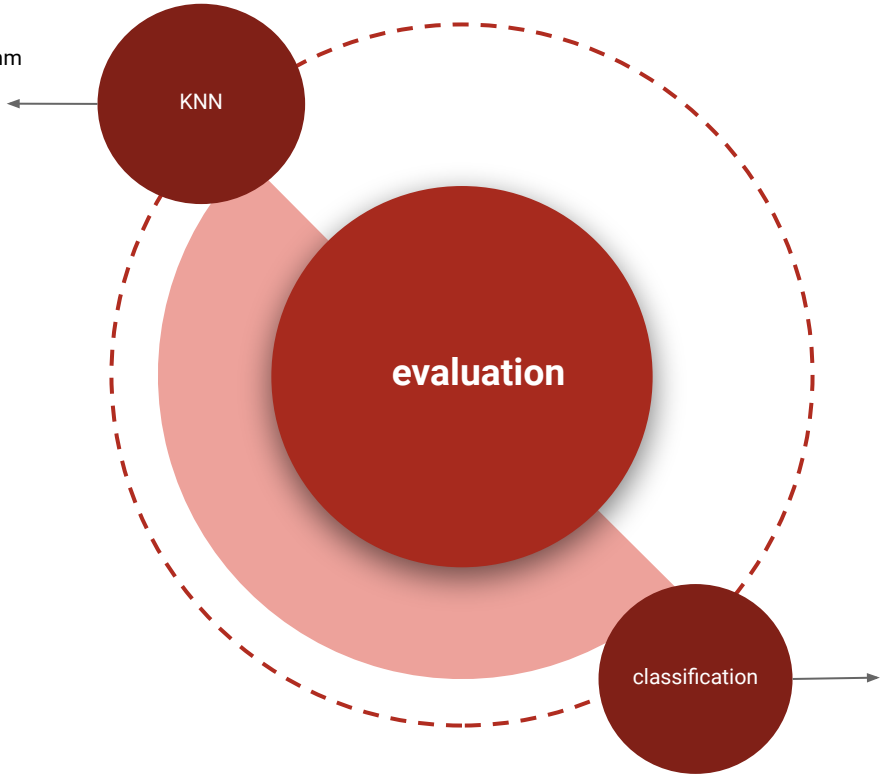
Use fully connected layer to build representation and make classification.



snnTorch PyTorch

Evaluation metric

Use representation in a KNN algorithm and observe the results



Add a fully connected layer to reduce dimension of representation for a classification task and observe the result and also compare with supervised learning for same network with less labeled data .

Road map

1 — 2 — 3 — 4

Spiking neural network

- introduction
- Spike encoding
- Neuron model
- Training SNN

Self-supervised learning

- Introduction
- Contrastive learning(simclr)
- Loss function

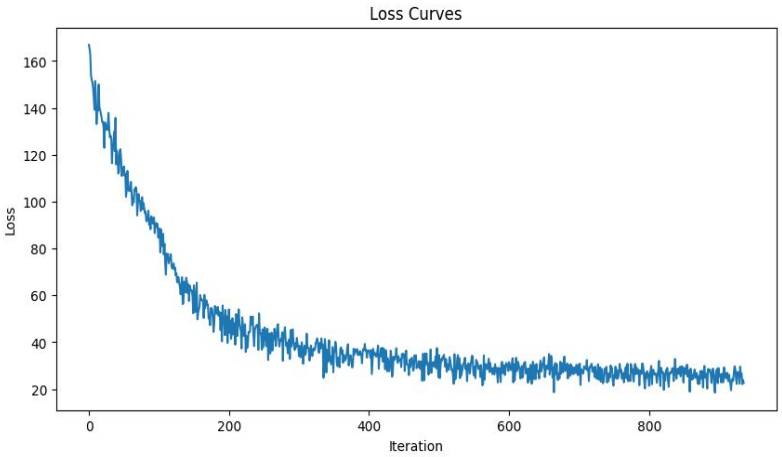
implementation

- Main idea
- Data augmentation
- Network architecture
- Evaluation metric

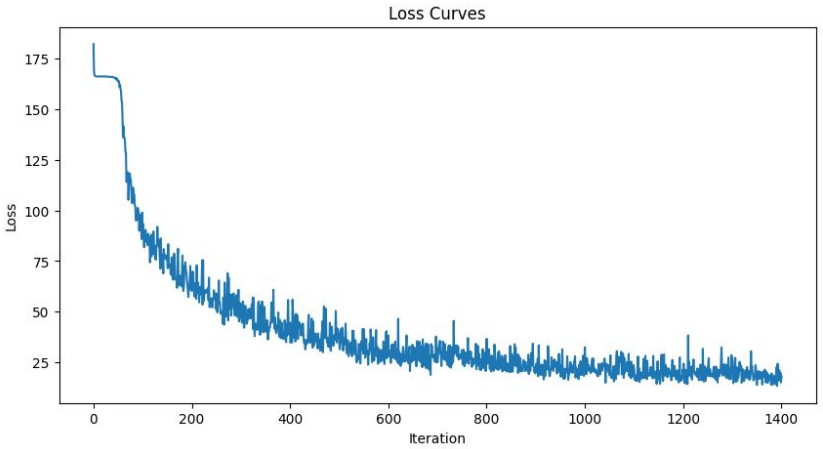
result

- Contrastive loss
- KNN
- classifier
- Research perspective

Contrastive loss

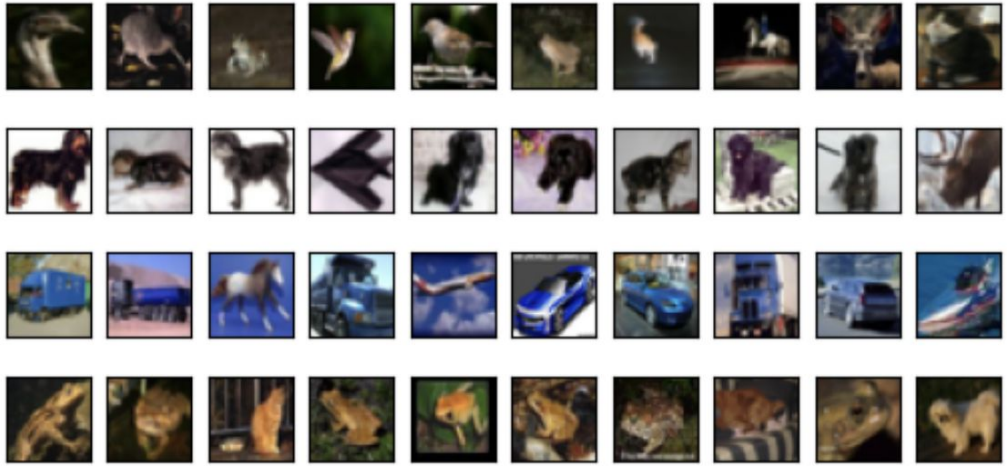


Loss function for simclr SNN on CIFAR10



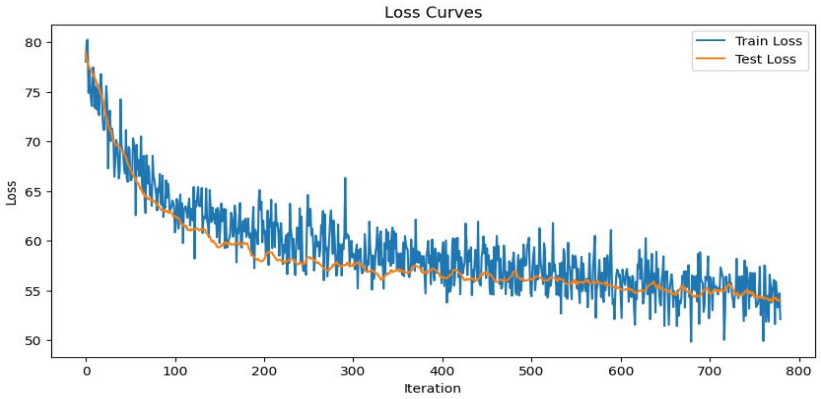
Loss function for simclr SNN on MNIST

KNN

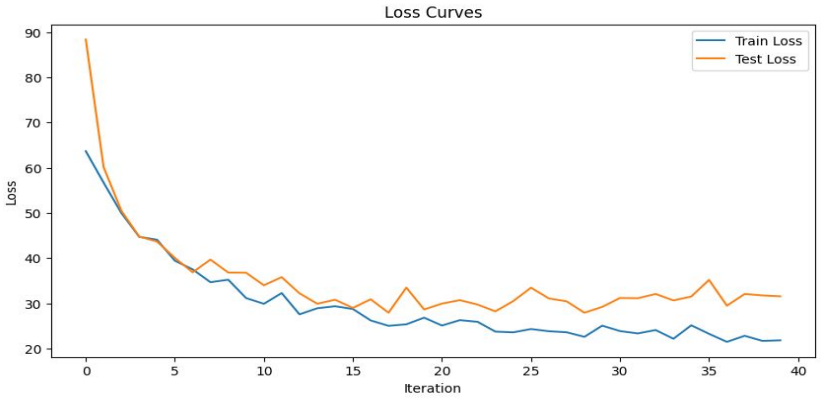


Samples of KNN algorithm implemented on representation from simclr SNN

Classifier

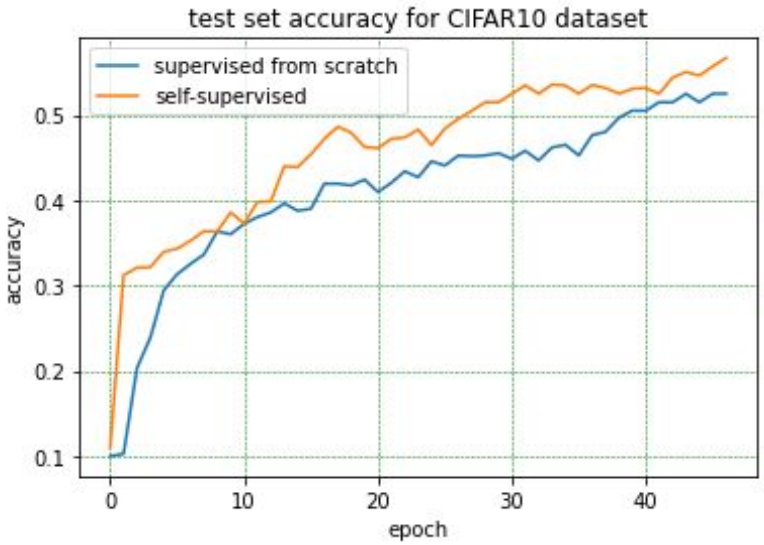


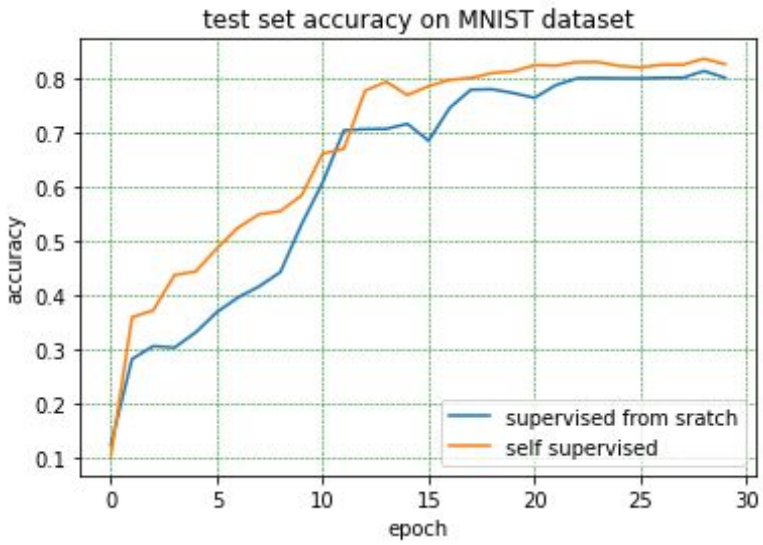
Loss function for classification CIFAR10 with 20% of labeled data



Loss function for classification MNIST with 20% of labeled data

Classifier


test set accuracy for CIFAR10 dataset
supervised from scratch
self-supervised
accuracy
epoch
0.1
0.2
0.3
0.4
0.5
0
10
20
30
40


test set accuracy on MNIST dataset
supervised from sratch
self supervised
accuracy
epoch
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
5
10
15
20
25
30

Research perspective

1. **Use more complex benchmarks**
   Use complex datasets for testing the network performance
2. **Deepen the network**
   Add layers and nodes to the network to do more complex tasks
3. **Different Spike encoding**
   Test different spike encoding for modifying network performance

Thank you for your time and attention