Midterm Project Proposal: SNN for Image Classification

Authors: SHENG-QIAN LI, JIA-JUN LAI, NAI-REN SONG, FANG-KAI

1. problem you want to solve and why

Table 4: Classification Accuracy for Poisson Encoded Input

Neuron Model	Learning Rule	Accuracy
IF	PostPre (STDP)	0.10
IF	WeightDependentPostPre	0.09
IF	Hebbian	0.10
LIF	PostPre (STDP)	0.10
LIF	WeightDependentPostPre	0.13
LIF	Hebbian	0.10
SRM0	PostPre (STDP)	0.10
SRM0	WeightDependentPostPre	0.10
SRM0	Hebbian	0.10
DiehlAndCook	PostPre (STDP)	0.79
DiehlAndCook	WeightDependentPostPre	0.79
DiehlAndCook	Hebbian	0.80

Figure 1 Poisson Encoded

Table 5: Classification Accuracy for Bernoulli Encoded Input

Neuron Model	Learning Rule	Accuracy
IF	PostPre (STDP)	0.10
IF	WeightDependentPostPre	0.10
IF	Hebbian	0.10
LIF	PostPre (STDP)	0.10
LIF	WeightDependentPostPre	0.10
LIF	Hebbian	0.10
SRM0	PostPre (STDP)	0.10
SRM0	WeightDependentPostPre	0.09
SRM0	Hebbian	0.10
DiehlAndCook	PostPre (STDP)	0.37
DiehlAndCook	WeightDependentPostPre	0.34
DiehlAndCook	Hebbian	0.35

Figure 2 Bernoulli Encoded

Table 6: Classification Accuracy for Rank Order Encoded Input

Neuron Model	Learning Rule	Accuracy
IF	PostPre (STDP)	0.10
IF	WeightDependentPostPre	0.10
IF	Hebbian	0.10
LIF	PostPre (STDP)	0.10
LIF	WeightDependentPostPre	0.10
LIF	Hebbian	0.10
SRM0	PostPre (STDP)	0.10
SRM0	WeightDependentPostPre	0.10
SRM0	Hebbian	0.10
DiehlAndCook	PostPre (STDP)	0.14
DiehlAndCook	WeightDependentPostPre	0.11
DiehlAndCook	Hebbian	0.13

Figure 3 Rank Order Encoded

The three pictures above show the accuracy results of the three encoding methods of Poisson Encoded, Bernoulli Encoded and Rank Order Encoded with different learning rules. It can be observed that the DichlAndCook model has the biggest difference under different encoding methods.

When the DiehlAndCook model uses Poisson Encoded Input, the accuracy can reach above 0.8 or even better. However, when using Bernoulli Encoded Input, the results are not as good as we imagined. The best Accuracy is 0.37 using PostPre.

Our goal is to improve the adaptability of the DiehlAndCook model to make it more suitable for different types of input encodings, thus improving its applicability to a wider range of tasks. In addition to solving the significant differences in the DiehlAndCook model under different encoded inputs, we also hope to improve the accuracy of other neuro models under different encoded inputs.

2. Technical part: how do you propose to solve it?

First, understand the differences between these three Learning Rules (PostPre (STDP), WeightDependentPostPre, Hebbian), differences between the models (IF, LIF, SRMO DiehlAndCook), and understand how many Neural Models and what algorithms are used.

In the second step, we search online to see if there are other Neural Models, and then use the same environment (batch size = 64, lr=1e-4, etc.), using MINST data and three Learning Rules (PostPre(STDP), conduct training under Hebbian, WeightDependentPostPre) and observe whether the final Accuracy is better or even higher than DiehlAndCook.

In addition to searching for other neuron models online, we can also adjust the hyperparameters to improve the performance of the model. The main adjustable hyperparameters are as shown in Figure 4:

Table 2: Spiking Neural Network Hyperparameters

Models	Hyperparameter	Value
All	Threshold Voltage	-52
All	Post-Spike Reset Voltage	-60
All	Refractory Period	5
LIF, SRM0, D&C	Membrane Potential Decay Constant	100
LIF, SRM0, D&C	Resting Voltage	-65
D&C	Threshold Voltage Increase (Adaptive)	0.05
D&C	Threshold Voltage Decay Constant	10^{-7}

Figure 4 adjustable hyperparameters

Finally, get a better model for accuracy from the second step, then understand the structure of its neuron models and compare it with the DiehlAndCook model. Through the comparison results, analyze why other Neural Models can improve Accuracy, and then try to change it in the DiehlAndCook model. algorithms, or add partial structures from other neuron models to improve accuracy.

3. Milestones achieved so far

We are currently implementing the DiehlAndCook model and comparing different encoding schemes and learning rules. The following figure shows several combinations with higher accuracy:

(1) Encoding scheme: Poisson

Leaning rule: PostPre

```
Encoding Scheme: Poisson
Neural Model: DiehlAndCook
Learning Technique: PostPre

Begin training.


All activity accuracy: 10.64 (last), 10.64 (average), 10.64 (best)
Proportion weighting accuracy: 10.64 (last), 10.64 (average), 10.64 (best)
Progress: 6400 / 60000

All activity accuracy: 64.25 (last), 37.45 (average), 64.25 (best)
Proportion weighting accuracy: 65.27 (last), 37.95 (average), 65.27 (best)
Progress: 12800 / 60000

All activity accuracy: 69.69 (last), 48.19 (average), 69.69 (best)
Proportion weighting accuracy: 71.58 (last), 49.16 (average), 71.58 (best)
Progress: 19200 / 60000

All activity accuracy: 83.75 (last), 57.08 (average), 83.75 (best)
Proportion weighting accuracy: 83.67 (last), 57.79 (average), 83.67 (best)
Progress: 25600 / 60000

All activity accuracy: 84.48 (last), 62.56 (average), 84.48 (best)
Proportion weighting accuracy: 84.19 (last), 63.07 (average), 84.19 (best)
Progress: 32000 / 60000

All activity accuracy: 82.56 (last), 65.90 (average), 84.48 (best)
Proportion weighting accuracy: 82.80 (last), 66.36 (average), 84.19 (best)
Progress: 38400 / 60000

All activity accuracy: 81.02 (last), 68.06 (average), 84.48 (best)
Proportion weighting accuracy: 81.83 (last), 68.57 (average), 84.19 (best)
Progress: 44800 / 60000

All activity accuracy: 80.66 (last), 69.63 (average), 84.48 (best)
Proportion weighting accuracy: 81.44 (last), 70.18 (average), 84.19 (best)
Progress: 51200 / 60000

All activity accuracy: 79.41 (last), 70.72 (average), 84.48 (best)
Proportion weighting accuracy: 80.09 (last), 71.28 (average), 84.19 (best)
Progress: 57600 / 60000
Training complete.
```

Figure 5 training accuracy

```
Begin testing


All activity accuracy: 0.81
Proportion weighting accuracy: 0.81

Testing complete.
```

Figure 6 testing accuracy

(2) Encoding scheme: Bernoulli

Leaning rule: PostPre

```
Encoding Scheme: Bernoulli
Neural Model: DiehlAndCook
Learning Technique: PostPre

Begin training.


All activity accuracy: 10.53 (last), 10.53 (average), 10.53 (best)
Proportion weighting accuracy: 10.53 (last), 10.53 (average), 10.53 (best)
Progress: 6400 / 60000

All activity accuracy: 10.00 (last), 10.27 (average), 10.53 (best)
Proportion weighting accuracy: 9.83 (last), 10.18 (average), 10.53 (best)
Progress: 12800 / 60000

All activity accuracy: 8.52 (last), 9.68 (average), 10.53 (best)
Proportion weighting accuracy: 9.09 (last), 9.82 (average), 10.53 (best)
Progress: 19200 / 60000

All activity accuracy: 12.69 (last), 10.43 (average), 12.69 (best)
Proportion weighting accuracy: 12.80 (last), 10.56 (average), 12.80 (best)
Progress: 25600 / 60000

All activity accuracy: 18.00 (last), 11.95 (average), 18.00 (best)
Proportion weighting accuracy: 18.20 (last), 12.09 (average), 18.20 (best)
Progress: 32000 / 60000

All activity accuracy: 28.03 (last), 14.63 (average), 28.03 (best)
Proportion weighting accuracy: 28.30 (last), 14.79 (average), 28.30 (best)
Progress: 38400 / 60000

All activity accuracy: 30.08 (last), 16.83 (average), 30.08 (best)
Proportion weighting accuracy: 30.95 (last), 17.10 (average), 30.95 (best)
Progress: 44800 / 60000

All activity accuracy: 33.84 (last), 18.96 (average), 33.84 (best)
Proportion weighting accuracy: 34.55 (last), 19.28 (average), 34.55 (best)
Progress: 51200 / 60000

All activity accuracy: 36.09 (last), 20.86 (average), 36.09 (best)
Proportion weighting accuracy: 36.88 (last), 21.24 (average), 36.88 (best)
Progress: 57600 / 60000
Training complete.
```

Figure 7 training accuracy

```
Begin testing

All activity accuracy: 0.39
Proportion weighting accuracy: 0.39

Testing complete.
```

Figure 8 testing accuracy

(3) Encoding scheme: RankOrder

Leaning rule: PostPre

```
Encoding Scheme: RankOrder
Neural Model: DiehlAndCook
Learning Technique: PostPre

Begin training.


All activity accuracy: 10.97 (last), 10.97 (average), 10.97 (best)
Proportion weighting accuracy: 10.97 (last), 10.97 (average), 10.97 (best)
Progress: 6400 / 60000

All activity accuracy: 16.17 (last), 13.57 (average), 16.17 (best)
Proportion weighting accuracy: 16.09 (last), 13.53 (average), 16.09 (best)
Progress: 12800 / 60000

All activity accuracy: 14.62 (last), 13.92 (average), 16.17 (best)
Proportion weighting accuracy: 14.62 (last), 13.90 (average), 16.09 (best)
Progress: 19200 / 60000

All activity accuracy: 14.53 (last), 14.07 (average), 16.17 (best)
Proportion weighting accuracy: 14.53 (last), 14.05 (average), 16.09 (best)
Progress: 25600 / 60000

All activity accuracy: 16.42 (last), 14.54 (average), 16.42 (best)
Proportion weighting accuracy: 16.42 (last), 14.53 (average), 16.42 (best)
Progress: 32000 / 60000

All activity accuracy: 15.14 (last), 14.64 (average), 16.42 (best)
Proportion weighting accuracy: 15.14 (last), 14.63 (average), 16.42 (best)
Progress: 38400 / 60000

All activity accuracy: 16.22 (last), 14.87 (average), 16.42 (best)
Proportion weighting accuracy: 16.22 (last), 14.86 (average), 16.42 (best)
Progress: 44800 / 60000

All activity accuracy: 14.98 (last), 14.88 (average), 16.42 (best)
Proportion weighting accuracy: 14.98 (last), 14.87 (average), 16.42 (best)
Progress: 51200 / 60000

All activity accuracy: 14.55 (last), 14.85 (average), 16.42 (best)
Proportion weighting accuracy: 14.55 (last), 14.84 (average), 16.42 (best)
Progress: 57600 / 60000
Training complete.
```

Figure 9 training accuracy

```
Begin testing

All activity accuracy: 0.13
Proportion weighting accuracy: 0.13

Testing complete.
```

Figure 10 testing accuracy

After installing the environment for implementation requirements, we started to copy some experiments in the paper and found that the overall data was roughly the same as the paper, which proved that we were in the same direction as the paper. During the training process, we found that RankOrder Encoded Input takes the most time to train, and Bernoulli is the fastest, and Bernoulli's accuracy is higher than RankOrder. This is different from the intuitive feeling of the model. The more complex the model, the longer it usually takes to train. However, more complex models tend to have better performance at the extreme limit of accuracy. Different combinations of encoded input and learning rules may bring about very ideal results. Both are equally important changes.

4. Remaining milestones (dates and sub-goals)

Different Combination of model and rule:

Use other encoding methods, learning rules, and SNN model combinations in bindsnet to improve accuracy, such as: Null Encoder, Repeat Encoder, MSTDP, Rmax, AdaptiveLIFNodes, etc.

Different Dataset:

Try applying the same SNN model to different datasets, such as CIFAR10, STL10, etc. to determine whether the dataset and model are inconsistent.

Hyperparameters:

The figure below is the information provided by the reference material. It is mentioned that the hyperparameters in the SNN model are adjustable.

Hyperparameters	Values
Timescale	[10, 20, 40, 50, 200]
Spatial data reduction	[full image (13 x 21), row-stride 13, row-stride 26, stride 21, stride 42, box 2 2, box 3 3, box 4 4, box 2 4, box 4 2, box 2 8, box 8 2 box 4 8, box 8 4]
pT cut-off	[0.2, 0.5, 0.7]
EONS fitness	[Accuracy, penalty, combination]
Bias	[True, False]

Figure 11 adjustable hyperparameters

Adjusting timescale, spatial data, bias, etc. to improve the training accuracy, SNN and CNN have different hyperparameters. Since SNN is a relatively new concept, it is necessary to understand how to adjust the hyperparameters to make the model perform better. This is also one of the research projects dedicated to this topic.