experiment I

parameters

```
deviation = 0.001
deviation_hold = 0.0005
```

outputs

```
sqlite version: 2.2.2
Slice length: 60
Training data:
long points: 1452
short points: 1453
hold points: 2680
Dataset has been saved to data/SPX_1m_TrainingData.pth.
DataProcessor for Training ======================================== Done.

Slice length: 60
Testing data:
long points: 353
short points: 354
hold points: 714
Dataset has been saved to data/SPX_1m_TestingData.pth.
DataProcessor for Testing ======================================== Done.

main() ================================ Done
Execution time of main(): 36.0318 seconds


learning_rate = 0.0001
num_epochs = 20
```

result:

```
Epoch 17
-------------------------------
loss: 4.224940  [   32/ 5585]
loss: 0.759045  [ 1056/ 5585]
loss: 1.014939  [ 2080/ 5585]
loss: 2.451941  [ 3104/ 5585]
loss: 0.780136  [ 4128/ 5585]
loss: 0.093659  [ 5152/ 5585]
Execution time of train(): 1.1695 seconds
Test result: Accuracy: 50.2%, Avg loss: 2.304163

Epoch 18
-------------------------------
loss: 4.207283  [   32/ 5585]
loss: 0.818694  [ 1056/ 5585]
loss: 1.204649  [ 2080/ 5585]
loss: 1.486126  [ 3104/ 5585]
loss: 1.008906  [ 4128/ 5585]
loss: 0.279065  [ 5152/ 5585]
Execution time of train(): 1.1678 seconds
Test result: Accuracy: 50.2%, Avg loss: 2.037825

Epoch 19
-------------------------------
loss: 3.310409  [   32/ 5585]
loss: 0.731746  [ 1056/ 5585]
loss: 1.002562  [ 2080/ 5585]
loss: 2.754482  [ 3104/ 5585]
loss: 0.975072  [ 4128/ 5585]
loss: 0.438903  [ 5152/ 5585]
Execution time of train(): 1.1592 seconds
Test result: Accuracy: 50.2%, Avg loss: 1.522703

Epoch 20
-------------------------------
loss: 3.242041  [   32/ 5585]
loss: 0.419936  [ 1056/ 5585]
loss: 1.213330  [ 2080/ 5585]
loss: 2.302600  [ 3104/ 5585]
loss: 0.935441  [ 4128/ 5585]
loss: 0.239978  [ 5152/ 5585]
```

```
Execution time of train(): 1.1964 seconds
Test result: Accuracy: 50.2%, Avg loss: 1.956514
```

Conclusion

> 👎 😨 loss value decrease, but accuracy keep the same from start epoch to the end.

experiment II

parameters

```
deviation = 0.002
deviation_hold = 0.001
```

outputs

```
sqlite version: 2.2.2
Slice length: 60
Training data:
long points: 654
short points: 654
hold points: 1597
Dataset has been saved to data/SPX_1m_TrainingData.pth.
DataProcessor for Training ======================================== Done.

Slice length: 60
Testing data:
long points: 158
short points: 158
hold points: 391
Dataset has been saved to data/SPX_1m_TestingData.pth.
DataProcessor for Testing ======================================== Done.

main() ================================ Done
Execution time of main(): 26.7242 seconds
```

results

```
Epoch 1
-------------------------------
loss: 1.152308  [   32/ 2905]
loss: 1.797088  [ 1056/ 2905]
loss: 0.889592  [ 2080/ 2905]
Execution time of train(): 0.5434 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.611442

Epoch 2
-------------------------------
loss: 3.796343  [   32/ 2905]
loss: 1.327879  [ 1056/ 2905]
loss: 0.911485  [ 2080/ 2905]
Execution time of train(): 0.5472 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.246782

Epoch 3
-------------------------------
loss: 2.801435  [   32/ 2905]
loss: 1.410393  [ 1056/ 2905]
loss: 0.902353  [ 2080/ 2905]
Execution time of train(): 0.5381 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.131667

Epoch 4
-------------------------------
loss: 2.304166  [   32/ 2905]
loss: 1.326275  [ 1056/ 2905]
loss: 0.834715  [ 2080/ 2905]
Execution time of train(): 0.5211 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.440944

Epoch 5
-------------------------------
loss: 3.264903  [   32/ 2905]
loss: 1.415622  [ 1056/ 2905]
loss: 0.924846  [ 2080/ 2905]
Execution time of train(): 0.5456 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.185645

Epoch 6
```

```
------------------------------
loss: 2.459006  [   32/ 2905]
loss: 1.383709  [ 1056/ 2905]
loss: 0.916479  [ 2080/ 2905]
Execution time of train(): 0.5322 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.545271

Epoch 7
------------------------------
loss: 3.678676  [   32/ 2905]
loss: 1.334977  [ 1056/ 2905]
loss: 0.961071  [ 2080/ 2905]
Execution time of train(): 0.5242 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.337583

Epoch 8
------------------------------
loss: 2.792360  [   32/ 2905]
loss: 1.399690  [ 1056/ 2905]
loss: 0.855236  [ 2080/ 2905]
Execution time of train(): 0.5196 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.639217

Epoch 9
------------------------------
loss: 3.662404  [   32/ 2905]
loss: 1.455803  [ 1056/ 2905]
loss: 0.953219  [ 2080/ 2905]
Execution time of train(): 0.5461 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.085197

Epoch 10
------------------------------
loss: 2.262254  [   32/ 2905]
loss: 1.404930  [ 1056/ 2905]
loss: 0.919327  [ 2080/ 2905]
Execution time of train(): 0.5286 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.084907

Epoch 11
------------------------------
loss: 2.240177  [   32/ 2905]
loss: 1.457605  [ 1056/ 2905]
```

```
loss: 0.884378  [ 2080/ 2905]
Execution time of train(): 0.5226 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.272106

Epoch 12
-------------------------------
loss: 2.662108  [   32/ 2905]
loss: 1.496191  [ 1056/ 2905]
loss: 0.923095  [ 2080/ 2905]
Execution time of train(): 0.5232 seconds
Test result: Accuracy: 55.3%, Avg loss: 0.995221

Epoch 13
-------------------------------
loss: 1.674209  [   32/ 2905]
loss: 1.341347  [ 1056/ 2905]
loss: 0.872505  [ 2080/ 2905]
Execution time of train(): 0.5299 seconds
Test result: Accuracy: 55.3%, Avg loss: 1.028752

Epoch 14
-------------------------------
loss: 1.940650  [   32/ 2905]
loss: 1.384272  [ 1056/ 2905]
loss: 0.868829  [ 2080/ 2905]
Execution time of train(): 0.5256 seconds
Test result: Accuracy: 55.7%, Avg loss: 0.947268

Epoch 15
-------------------------------
loss: 1.436841  [   32/ 2905]
loss: 1.277346  [ 1056/ 2905]
loss: 0.839181  [ 2080/ 2905]
Execution time of train(): 0.5385 seconds
Test result: Accuracy: 55.7%, Avg loss: 0.917677

Epoch 16
-------------------------------
loss: 1.384798  [   32/ 2905]
loss: 1.300195  [ 1056/ 2905]
loss: 0.777339  [ 2080/ 2905]
Execution time of train(): 0.5463 seconds
Test result: Accuracy: 56.3%, Avg loss: 1.027306
```

```
Epoch 17
-------------------------------
loss: 1.855055  [   32/ 2905]
loss: 1.331372  [ 1056/ 2905]
loss: 0.785759  [ 2080/ 2905]
Execution time of train(): 0.5845 seconds
Test result: Accuracy: 56.0%, Avg loss: 1.009138

Epoch 18
-------------------------------
loss: 1.895379  [   32/ 2905]
loss: 1.354670  [ 1056/ 2905]
loss: 0.800765  [ 2080/ 2905]
Execution time of train(): 0.5353 seconds
Test result: Accuracy: 56.2%, Avg loss: 0.915927

Epoch 19
-------------------------------
loss: 1.546647  [   32/ 2905]
loss: 1.329480  [ 1056/ 2905]
loss: 0.759709  [ 2080/ 2905]
Execution time of train(): 0.5490 seconds
Test result: Accuracy: 56.3%, Avg loss: 0.932919

Epoch 20
-------------------------------
loss: 1.586259  [   32/ 2905]
loss: 1.299095  [ 1056/ 2905]
loss: 0.758787  [ 2080/ 2905]
Execution time of train(): 0.5260 seconds
Test result: Accuracy: 56.0%, Avg loss: 0.833086
```

Conclusion

💡 👉 little improved, which means the selection of peaks/troughs and holds are sensitive to the final Accuracy.

experiment III

parameters

```
deviation = 0.002
deviation_hold = 0.0018
```

outputs

```
sqlite version: 2.2.2
Slice length: 60
Training data:
long points: 654
short points: 654
hold points: 178
Dataset has been saved to data/SPX_1m_TrainingData.pth.
DataProcessor for Training ======================================== Done.

Slice length: 60
Testing data:
long points: 158
short points: 158
hold points: 52
Dataset has been saved to data/SPX_1m_TestingData.pth.
DataProcessor for Testing ======================================== Done.

main() ================================ Done
Execution time of main(): 21.2885 seconds
```

model environments

- parameters

```
learning_rate = 0.0001
num_epochs = 20
```

- data structure

```
Training data size: 1486, (torch.Size([60, 9]), torch.Size([3]))
Testing data size: 368, (torch.Size([60, 9]), torch.Size([1]))
```

- model class
-

```
  def main(self):
      self.loadData()
      self.defineModel("linear")
      self.train_test()
      self.save()
```

results

```
Epoch 1
-------------------------------
loss: 1.057150  [   32/ 1486]
loss: 1.836876  [ 1056/ 1486]
Execution time of train(): 0.2846 seconds
Test result: Accuracy: 47.8%, Avg loss: 1.003603

Epoch 2
-------------------------------
loss: 0.943238  [   32/ 1486]
loss: 0.968719  [ 1056/ 1486]
Execution time of train(): 0.2734 seconds
Test result: Accuracy: 47.0%, Avg loss: 0.970266

Epoch 3
-------------------------------
loss: 0.954880  [   32/ 1486]
loss: 0.886056  [ 1056/ 1486]
Execution time of train(): 0.2716 seconds
Test result: Accuracy: 49.2%, Avg loss: 0.947095

Epoch 4
-------------------------------
loss: 0.971180  [   32/ 1486]
loss: 0.854778  [ 1056/ 1486]
Execution time of train(): 0.2988 seconds
Test result: Accuracy: 53.8%, Avg loss: 0.902409

Epoch 5
-------------------------------
loss: 0.969154  [   32/ 1486]
loss: 0.812290  [ 1056/ 1486]
Execution time of train(): 0.2721 seconds
Test result: Accuracy: 56.2%, Avg loss: 0.878211

Epoch 6
-------------------------------
loss: 1.016719  [   32/ 1486]
loss: 0.781297  [ 1056/ 1486]
Execution time of train(): 0.2801 seconds
Test result: Accuracy: 62.2%, Avg loss: 0.807119
```

```
Epoch 7
-------------------------------
loss: 0.975887  [   32/ 1486]
loss: 0.689459  [ 1056/ 1486]
Execution time of train(): 0.2741 seconds
Test result: Accuracy: 64.9%, Avg loss: 0.767535

Epoch 8
-------------------------------
loss: 0.993949  [   32/ 1486]
loss: 0.653907  [ 1056/ 1486]
Execution time of train(): 0.2726 seconds
Test result: Accuracy: 71.5%, Avg loss: 0.694774

Epoch 9
-------------------------------
loss: 0.845538  [   32/ 1486]
loss: 0.566895  [ 1056/ 1486]
Execution time of train(): 0.2716 seconds
Test result: Accuracy: 74.2%, Avg loss: 0.633703

Epoch 10
-------------------------------
loss: 0.759297  [   32/ 1486]
loss: 0.489004  [ 1056/ 1486]
Execution time of train(): 0.2716 seconds
Test result: Accuracy: 75.8%, Avg loss: 0.623743

Epoch 11
-------------------------------
loss: 0.718356  [   32/ 1486]
loss: 0.455563  [ 1056/ 1486]
Execution time of train(): 0.2720 seconds
Test result: Accuracy: 78.3%, Avg loss: 0.561403

Epoch 12
-------------------------------
loss: 0.635166  [   32/ 1486]
loss: 0.402290  [ 1056/ 1486]
Execution time of train(): 0.2746 seconds
Test result: Accuracy: 79.6%, Avg loss: 0.533978
```

```
Epoch 13
-------------------------------
loss: 0.568633  [   32/ 1486]
loss: 0.374445  [ 1056/ 1486]
Execution time of train(): 0.2699 seconds
Test result: Accuracy: 81.2%, Avg loss: 0.528032

Epoch 14
-------------------------------
loss: 0.503371  [   32/ 1486]
loss: 0.344814  [ 1056/ 1486]
Execution time of train(): 0.2716 seconds
Test result: Accuracy: 82.3%, Avg loss: 0.506276

Epoch 15
-------------------------------
loss: 0.465778  [   32/ 1486]
loss: 0.321299  [ 1056/ 1486]
Execution time of train(): 0.2770 seconds
Test result: Accuracy: 83.2%, Avg loss: 0.483133

Epoch 16
-------------------------------
loss: 0.441938  [   32/ 1486]
loss: 0.303850  [ 1056/ 1486]
Execution time of train(): 0.2726 seconds
Test result: Accuracy: 83.7%, Avg loss: 0.477422

Epoch 17
-------------------------------
loss: 0.421485  [   32/ 1486]
loss: 0.290050  [ 1056/ 1486]
Execution time of train(): 0.2793 seconds
Test result: Accuracy: 83.7%, Avg loss: 0.480852

Epoch 18
-------------------------------
loss: 0.405973  [   32/ 1486]
loss: 0.279391  [ 1056/ 1486]
Execution time of train(): 0.2736 seconds
Test result: Accuracy: 84.0%, Avg loss: 0.457794

Epoch 19
```

```
------------------------------
loss: 0.388468  [   32/ 1486]
loss: 0.268329  [ 1056/ 1486]
Execution time of train(): 0.2736 seconds
Test result: Accuracy: 84.0%, Avg loss: 0.454665

Epoch 20
------------------------------
loss: 0.372735  [   32/ 1486]
loss: 0.258871  [ 1056/ 1486]
Execution time of train(): 0.2746 seconds
Test result: Accuracy: 84.5%, Avg loss: 0.447803
```

Conclusion

> 👍😄 Obviously, the accuracy has reached a good level, but there's potential to improve it further by fine-tuning the zigzag deviation and using different AI models.