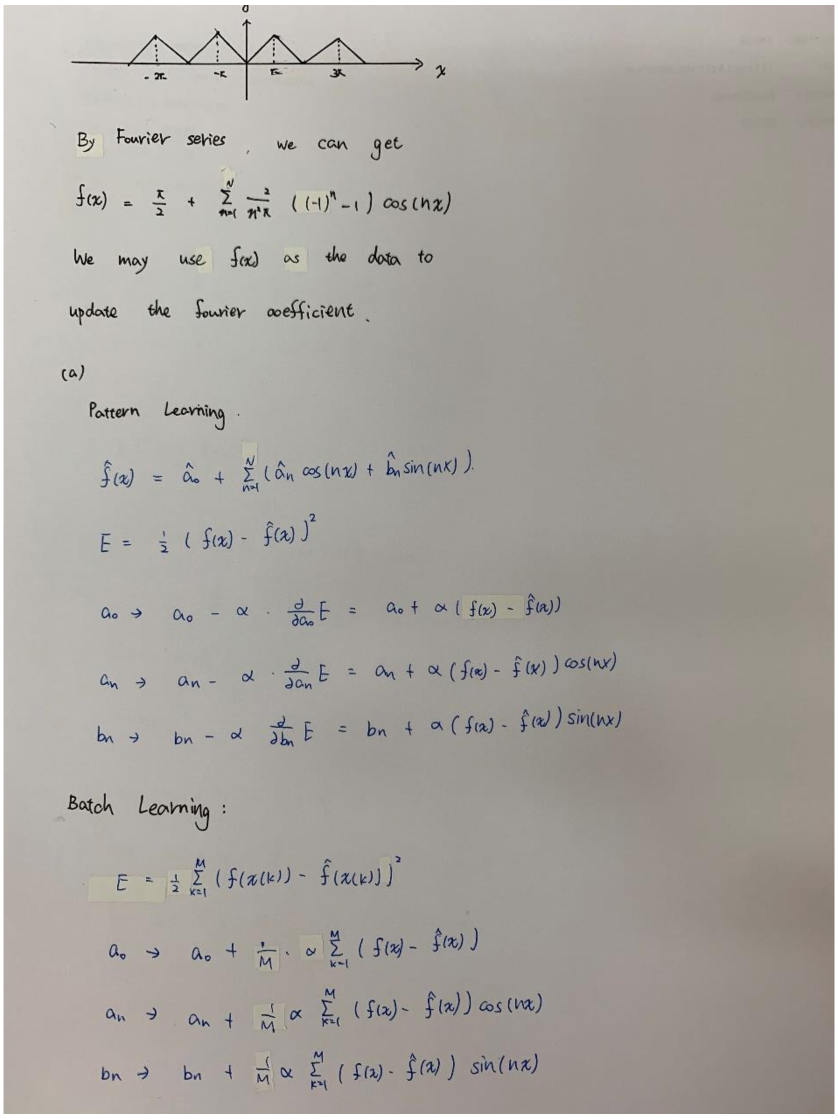
309512018 林靖凱

(a)



-2π $-\pi$ π 3π x

By Fourier series , we can get

$$f(x) = \frac{\pi}{2} + \sum_{n=1}^{N} \frac{2}{n^2\pi}\left((-1)^n - 1\right)\cos(nx)$$

We may use $f(x)$ as the data to update the fourier coefficient .

(a)

Pattern Learning .

$$\hat{f}(x) = \hat{a}_0 + \sum_{n=1}^{N}\left(\hat{a}_n \cos(nx) + \hat{b}_n \sin(nx)\right).$$

$$E = \frac{1}{2}\left(f(x) - \hat{f}(x)\right)^2$$

$$a_0 \rightarrow a_0 - \alpha \cdot \frac{\partial}{\partial a_0}E = a_0 + \alpha\left(f(x) - \hat{f}(x)\right)$$

$$a_n \rightarrow a_n - \alpha \cdot \frac{\partial}{\partial a_n}E = a_n + \alpha\left(f(x) - \hat{f}(x)\right)\cos(nx)$$

$$b_n \rightarrow b_n - \alpha \frac{\partial}{\partial b_n}E = b_n + \alpha\left(f(x) - \hat{f}(x)\right)\sin(nx)$$

Batch Learning :

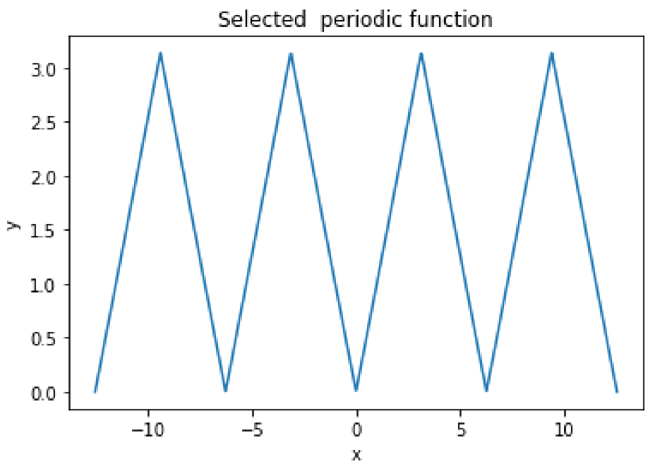
$$E = \frac{1}{2}\sum_{k=1}^{M}\left(f(x(k)) - \hat{f}(x(k))\right)^2$$

$$a_0 \rightarrow a_0 + \frac{1}{M} \cdot \alpha \sum_{k=1}^{M}\left(f(x) - \hat{f}(x)\right)$$

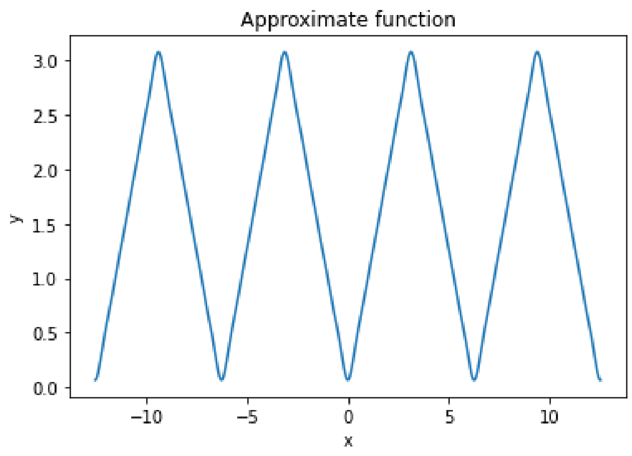
$$a_n \rightarrow a_n + \frac{1}{M}\alpha \sum_{k=1}^{M}\left(f(x) - \hat{f}(x)\right)\cos(nx)$$

$$b_n \rightarrow b_n + \frac{1}{M}\alpha \sum_{k=1}^{M}\left(f(x) - \hat{f}(x)\right)\sin(nx)$$

Define function as follow and the data is chosen 1000 by uniformly choose:

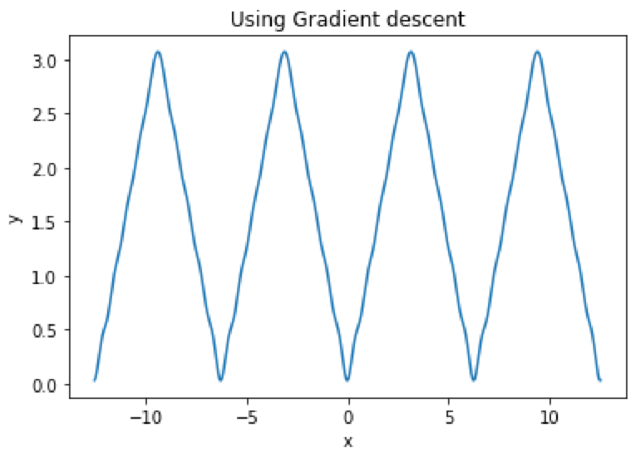


And using N = 10 to calculate the Fourier series to approximate it as follows:



(b)

Using learning rate = 0.01 and initial value a0 = π, an = 0.5 and bn = 0.5 to update the Fourier coefficient and we can get the MSE is equal to 0.000543 and the function shown as follow



```
MSE = 0.000543
```

Below figure shows the coefficients a0, an, bn of Fourier series and gradient descent to update coefficient:

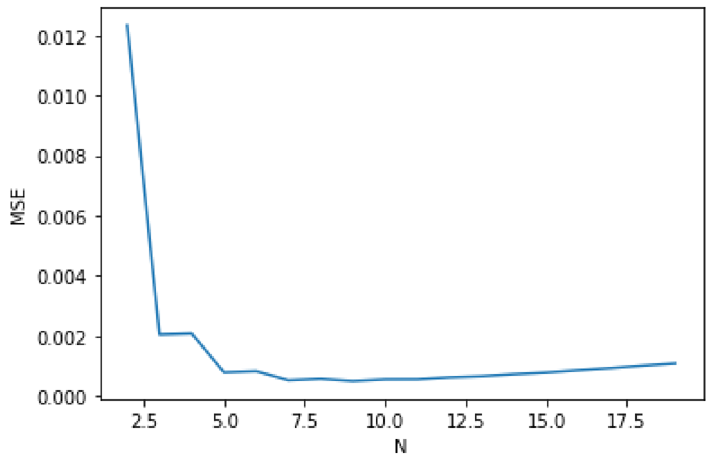
```
===============================
=== Approximate coefficient ===
a0 = 1.570796
an = [-1.2732395447351628, 0.0, -0.1414710605261292, 0.0, -0.05092958178940651, 0.0, -0.025984480504799924, 0.0, -0.015719006
725125467]
bn = [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
===============================
===============================
==== GD update coefficient ====
a0 = 1.575520
an = [-1.27216972e+00  2.25298370e-03 -1.39753030e-01  1.09388282e-03
 -5.08033040e-02 -9.45481368e-04 -2.87476975e-02 -5.13113275e-03
 -2.56788608e-02 -2.16566418e-02]
bn = [0.00488058 0.00099759 0.00246256 0.0024983  0.00338417 0.00359785
 0.00431339 0.00447822 0.00512562 0.00601139]
===============================
```

We can notice the coefficient of two way are a little different which are very close and the difference is about three decimal places. Gradient descent to update the coefficient can get the roughly the same function, if the N is big enough.

(c)

Using different order N and learning rate is equal to 0.01 to observe the Mean-square-error and the order N is from 2 to 20, the figure shows as below:

```
MSE = 0.012342
MSE = 0.002044
MSE = 0.002077
MSE = 0.000778
MSE = 0.000817
MSE = 0.000514
MSE = 0.000562
MSE = 0.000488
MSE = 0.000543
MSE = 0.000546
MSE = 0.000609
MSE = 0.000646
MSE = 0.000716
MSE = 0.000772
MSE = 0.000848
MSE = 0.000917
MSE = 0.001000
MSE = 0.001079
```

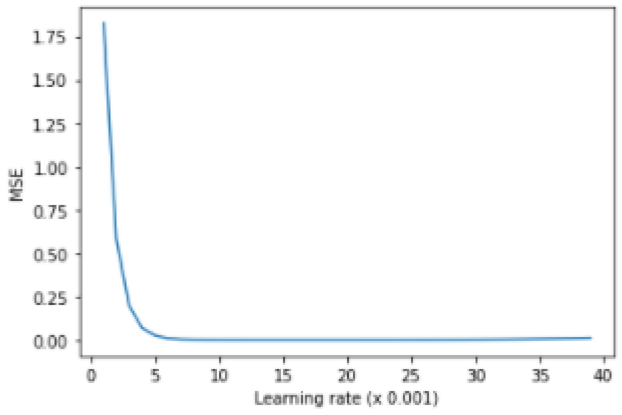


We can notice that the beginning N = 2 that the MSE is very big and when N approach to 9 the MSE has the minimum value, but when N rises, the MSE has the growing trend.

(d)

Using different order learning rate and order N is equal to 10 to observe the Mean-square-error and the learning rate is from 0.001 to 0.04, the figure shows as below:

```
MSE = 1.827205
MSE = 0.581466
MSE = 0.198647
MSE = 0.070485
MSE = 0.025725
MSE = 0.009700
MSE = 0.003848
MSE = 0.001672
MSE = 0.000852
MSE = 0.000543
MSE = 0.000430
MSE = 0.000391
MSE = 0.000378
MSE = 0.000375
MSE = 0.000373
MSE = 0.000370
MSE = 0.000368
MSE = 0.000366
MSE = 0.000366
MSE = 0.000371
MSE = 0.000384
MSE = 0.000409
MSE = 0.000454
MSE = 0.000525
MSE = 0.000632
MSE = 0.000784
MSE = 0.000992
MSE = 0.001269
MSE = 0.001627
MSE = 0.002078
MSE = 0.002636
MSE = 0.003313
MSE = 0.004121
MSE = 0.005072
MSE = 0.006178
MSE = 0.007450
MSE = 0.008897
MSE = 0.010529
MSE = 0.012355
```



We can notice that the MSE is very big that is because that the learning rate is so small that it converges so slow. And when learning rate become larger, the MSE has a little growing trend, but not obvious. I think when learning rate become more and more large, the MSE will become large obviously. And in this part, we can notice that the learning rate is equal to 0.018 has the smallest MSE.

(e)

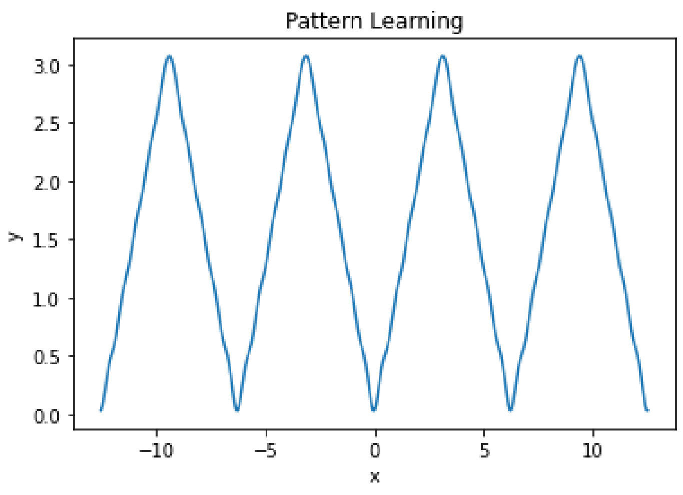
Pattern learning:

It takes all data to let computer learn the coefficients one by one.

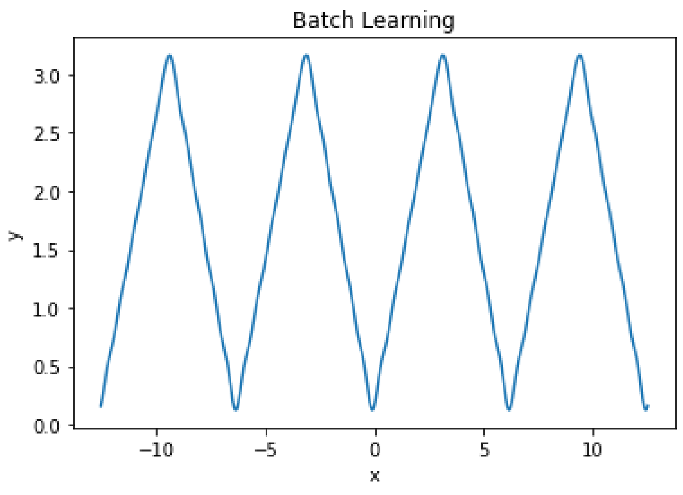
Batch learning:

It divides the data into N and update the coefficient by data / N.

In this part, I choose batch size is 5, learning rate is 0.01 and order N is 10. And the MSE of Batch learning is 0.011867. We can notice that the Pattern learning has more small value than Batch learning. I think the Pattern learning is more accurate than Batch learning because Pattern learning update coefficient every time.



```
Pattern learning MSE = 0.000543
```

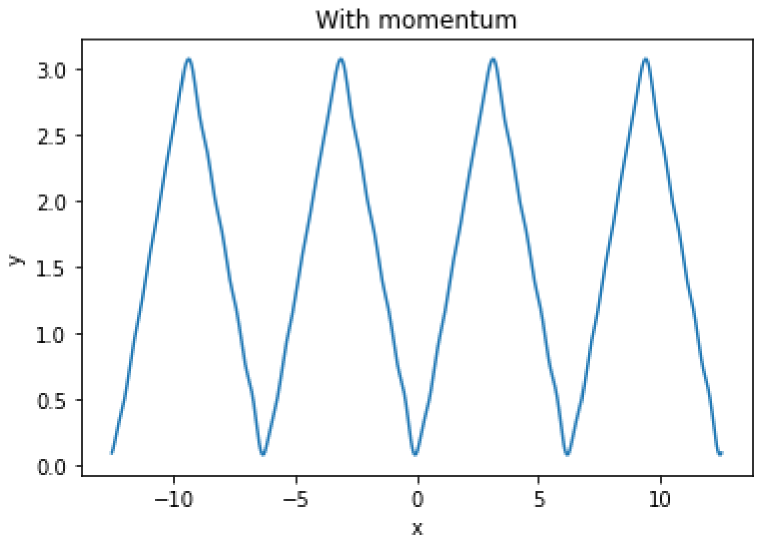


```
Batch learning MSE = 0.011867
```

(f)

In this part, I use order N = 10, learning rate is 0.01 and momentum is 0.1.

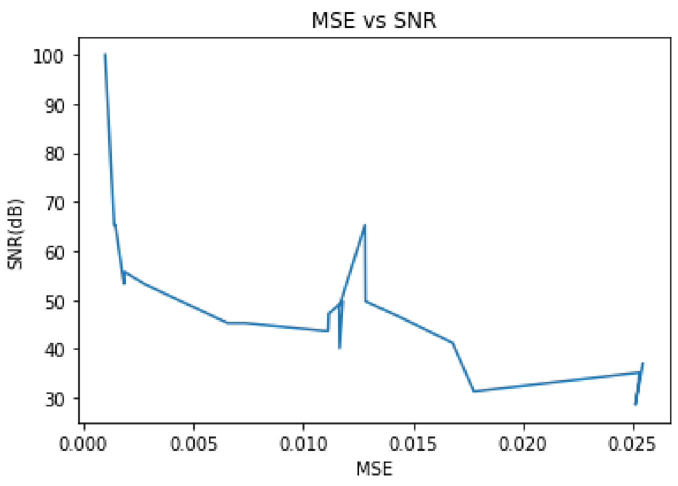
Gradient descent method with momentum term is used to prevent the local minimum and the concept of Momentum is as follows: when a ball rolls from a slope to flat ground, the ball will continue to roll on flat ground because the ball has momentum, that is, its speed is related to the speed at a point in time. But in this part, I think the local minimum is rarely occur, so the MSE of Gradient descent method without momentum has the smallest value. And the MSE of two ways has little difference.



```
MSE = 0.000626
```

(g)

In this part, we want to the relation of MSE and SNR. The noise is made randomly, learning rate is 0.01 and order N is 10. We can notice that when MSE become large that SNR will become which means that it has poor anti-noise ability. And the jump points, I think it is error by the randomly selected. Overall, the ability of Confrontational noise is acceptable.



(h)

I think it can be viewed as f(x) is approximated by linear combination of sinusoidal basis because MSE is acceptable. And polynomial basis I think is that it also can be used because polynomial basis is Orthogonal basis it can approximate any function, but it has large amount of calculation.