Machine Learning 2021 Homework 2 電機4C 洪愷尹 0710851

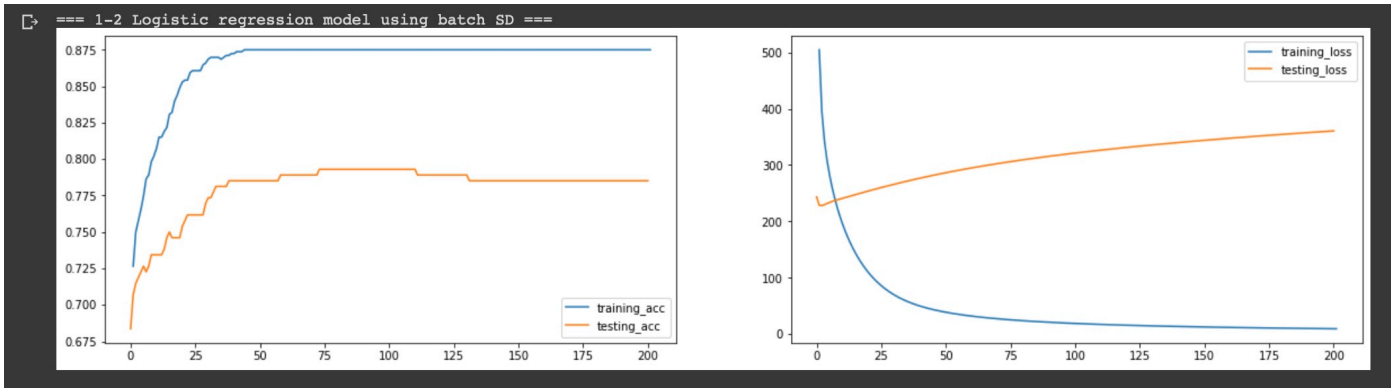
1. Classification Problem

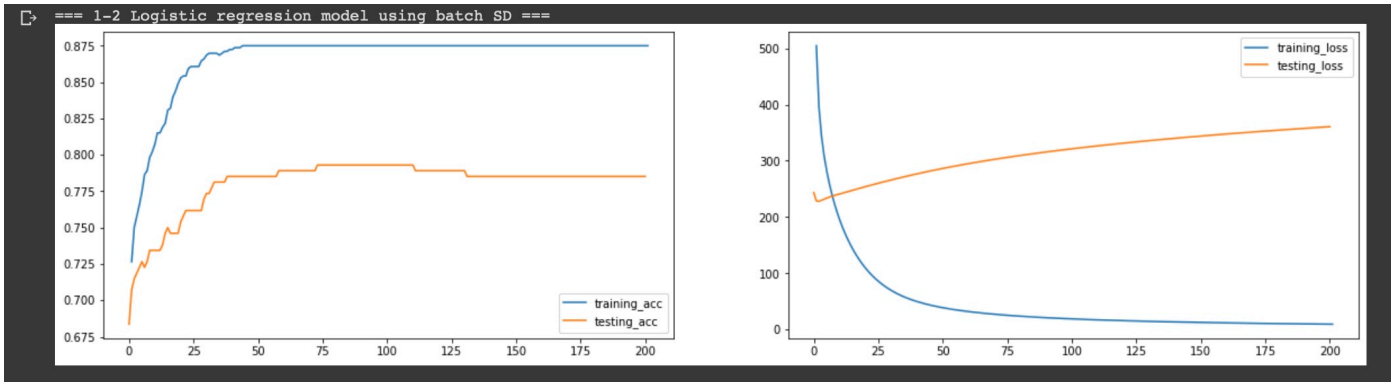
1-1

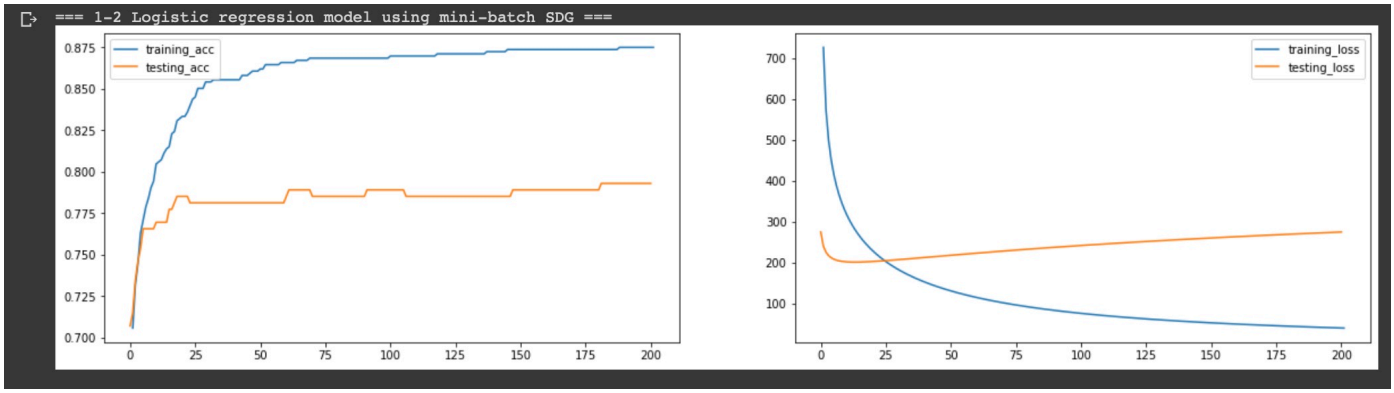
```
print("=== 1-1 Least squares for classification Problem ===")
print("- Training Dataset")
print("-- MSE loss : ", train_mse)
print("-- Accuracy : ", train_acc)
print("- Test Dataset")
print("-- MSE loss : ", test_mse)
print("-- Accuracy : ", test_acc)
```

```
=== 1-1 Least squares for classification Problem ===
- Training Dataset
-- MSE loss :  0.8025812610920604
-- Accuracy :  0.875
- Test Dataset
-- MSE loss :  31.23676469752535
-- Accuracy :  0.61328125
```

1-2-a







1-2-b

```
=== 1-2 Final Classification Accuracy and Loss ===

=== batch GD ===
[1]Training Dataset
 - MSE loss :  4.822960139089048
 - Accuracy :  87.36979166666666 %
[2]Test Dataset
 - MSE loss :  1744.4733213240293
 - Accuracy :  79.6875 %

=== SGD ===
[1]Training Dataset
 - MSE loss :  9.582723570554009
 - Accuracy :  87.5 %
[2]Test Dataset
 - MSE loss :  360.6889234081183
 - Accuracy :  78.515625 %

=== mini-batch SGD ===
[1]Training Dataset
 - MSE loss :  40.51798080643124
 - Accuracy :  87.5 %
[2]Test Dataset
 - MSE loss :  275.5368227294773
 - Accuracy :  79.296875 %
```

1-2-c

```
=== 1-2 batch GD / SGD / mini-batch SGD ===
在訓練200 epoch後，三個方法在training的的accuracy則是收斂在相似處。
Training過程中，batch GD理論上要是最平滑，SDG更新時理論上要是最隨機混論的
但圖顯然不是這樣表達的，我認為是Data的分佈不夠隨機造成的
```

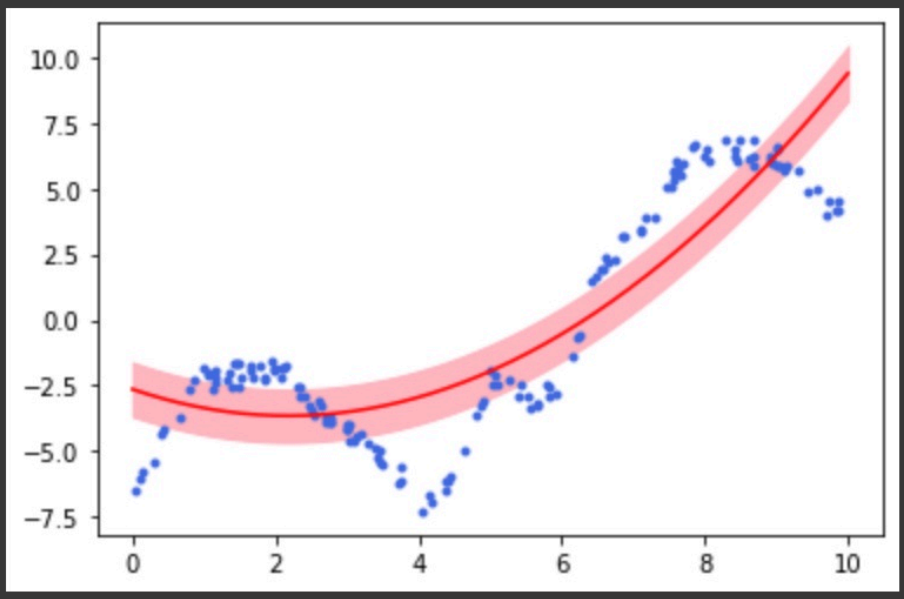
```
=== 1-3 Should we use the least squares model for classification problem ? ===
My answer would be yes, but LS's result may not do well, least-squares solutions lack robustness to outliers.
We can also observed from our experient that the testing result is really bad though training isn't !!
Besides, logistic way works better on testing dataset as we can found on above results.
Testing results in three different batchsize all show accuracy to some extent!!
```

---

2. Gaussian Process for Regression

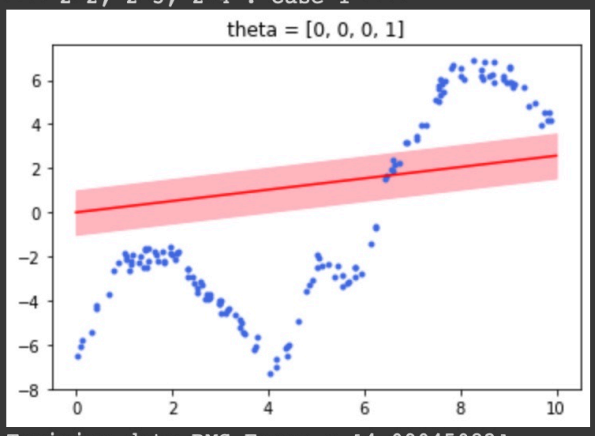
2-1

```
=== 2-1 ===
Training data RMS Error : [2.04592921]
Testing data RMS Error : [2.09657972]
```



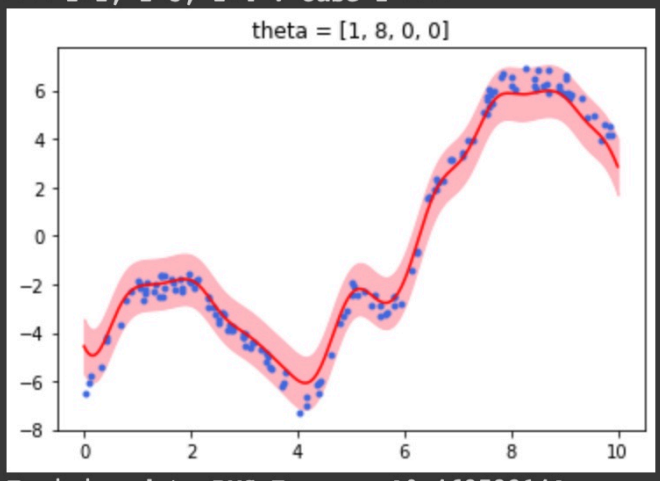
2-2 2-3 2-4

```
=== 2-2, 2-3, 2-4 : case 1 ===
```



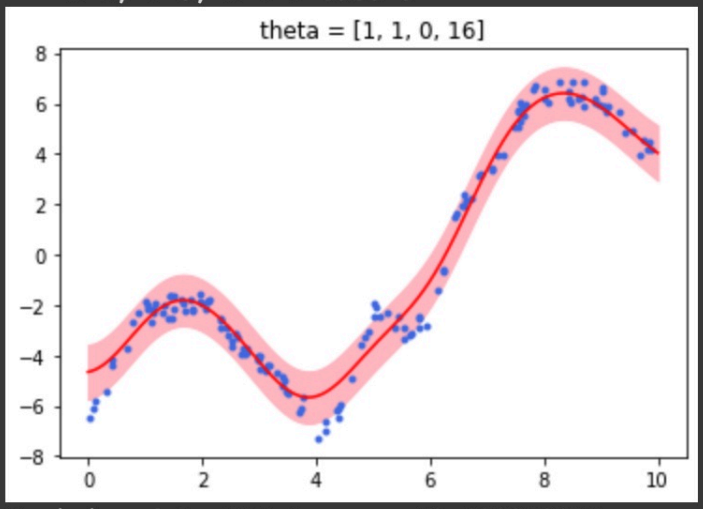

```
Training data RMS Error : [4.09045082]
Testing data RMS Error : [3.93528021]
```

```
=== 2-2, 2-3, 2-4 : case 2 ===
```



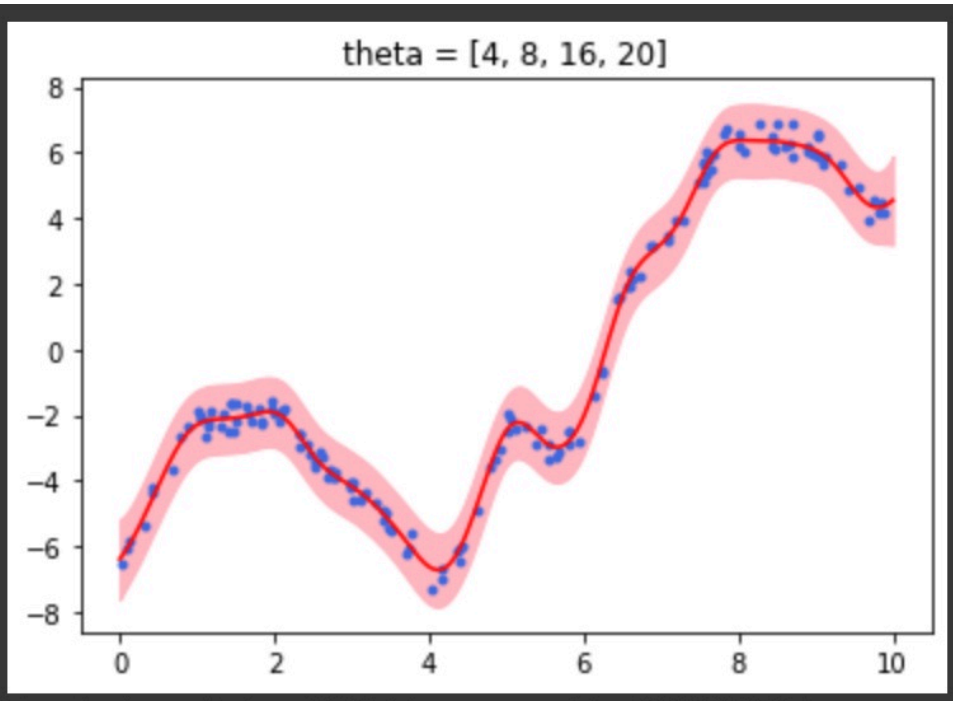

```
Training data RMS Error : [0.46873214]
Testing data RMS Error : [0.49366074]
```

```
=== 2-2, 2-3, 2-4 : case 3 ===
```




```
Training data RMS Error : [0.60287327]
Testing data RMS Error : [0.58307623]
```

2-5 Trial & Error To Find Hyperparameter




```
Training data RMS Error : [0.26219509]
Testing data RMS Error : [0.32787754]
```

2-6

```
=== 2-6 Discussion ===
[1] Discussion
- theta 0 : overall variance
- theta 1 : 1/lengthscale^2
- theta 2 : constant
- theaa 3 : linear constant
Trial & Error的時候，基於上述的參數去調整，可以利用高斯smooth點的分佈，但刻意調到到fit training data
會讓他線條不夠smooth。

[2] Discussion
將Expo跟Poly kernel比較可以看到，M=2的多項式Kernel，沒辦法General的表達Data的分佈，可見Model的複雜度不夠高
```