* **Execution description：**

在Linear\_Regression.py所在的資料夾中開啟terminal，輸入要執行的程式和dataset的指令，例如要用Linear\_Regression執行data\_1.txt，則輸入：python Linear\_Regression.py --path ..\2025\_CCU\_ML\_HW3\_Data\data\_1.txt，其中須確保DataLoader.py與此程式在相同資料夾下。

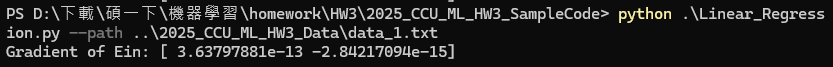
程式收到指令後開始讀取輸入的dataset是否存在，若不存在，則輸出File not found, please try again；若存在，則透過DataLoader讀取dataset送入Linear\_Regression函數執行。

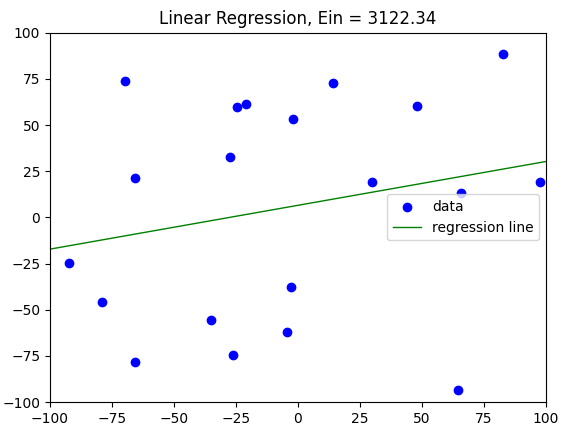
接著寫入程式，根據講義上的公式：進行計算最優weights，以及進行error的計算，最後再計算得到gradient值，到這function結束，回到main function繪出regression line。

* **Experimental results：**

1. 回歸直線：

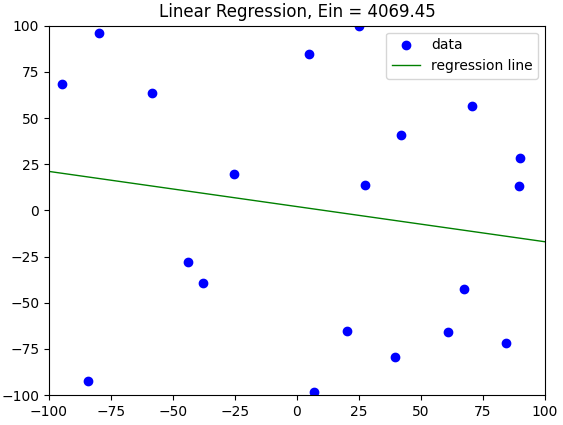
Data\_1.txt的結果





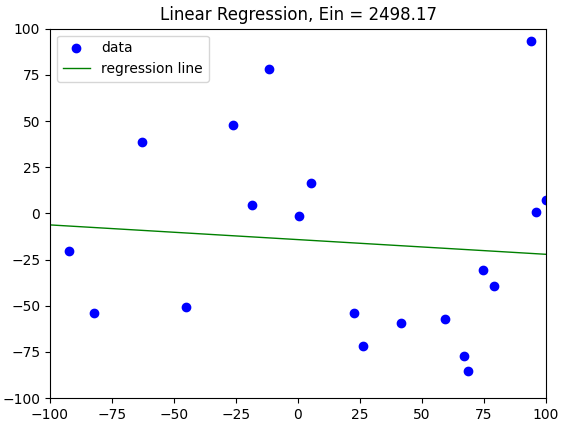
Data\_2.txt的結果





Data\_3.txt的結果





1. In-sample error

Data\_1.txt為3122.34

Data\_2.txt為4069.45

Data\_3.txt為2498.17

1. Gradient

Data\_1.txt為[ 3.63797881e-13 -2.84217094e-15]

Data\_2.txt為[-1.81898940e-13 1.11022302e-15]

Data\_3.txt為[0.00000000e+00 5.68434189e-15]

* **Conclusion：**

經由輸出可以發現兩個元素非常接近0，對應到講義ML\_06\_Linear\_Regression\_T的p.12中我們的任務是找到gradient值=0的時刻，這個時刻的weights也就是我們想找的regression line的參數，因此可以確定這個weights及regression line就是我們的答案，我們就weights return給main function即可。

* **Discussion：**

實作完function後，想說error怎麼那麼大，是不是寫錯了，還因此去問助教是否怪怪的，問的同時繼續把gradient部分寫完，並把他印出來後得到非常接近0的數字，就可以得知應該沒有寫錯。除此之外，也從助教那邊得知error那麼大是正常的，因為range設定為-100~100，導致差值的平方會很大。