**數值最佳化 期中作業**

組長: 工資112 吳旻恩

組員: 工資113 張柏駿 蘇奕幃 邱宇辰

**第一題：**

**(a)**

**凸集（Convex set）**是一個點集合，其中每兩點之間的線段點都落在該點集合中。若下列條件成立：給定任意 c1,c2∈C 且λ∈[0,1] 則 λc1+(1−λ)c2∈C

我們可以定義一個集合C⊂ Rk為 凸集合 ( convex set )

Examples:

一張含有 文字, 體育競賽, 運動 的圖片

自動產生的描述

1. 第一個圖是一個正六邊形，它是convex的，因爲它內部的任意兩點的連線也被包含在該圖形的內部。
2. 第二個圖不是convex的，其為凹集，因爲當我們連接set中的兩點時，出現了未被包含在set中的點，即上圖中紅色方框中的區域。
3. 第三個圖不是convex的，原因同圖二。

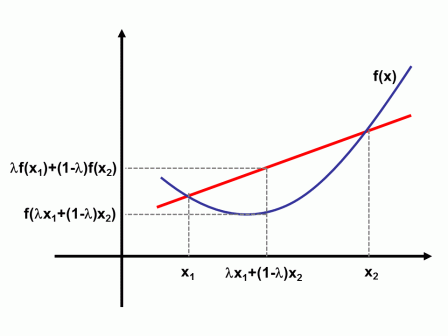
**凸函數（Convex function）**是令 K\subset\mathbb{R}^n 為一個非空凸集，也就是說，給定任意兩點 \mathbf{x}_1,\mathbf{x}_2\in K 和 0\le\lambda\le 1，點 \lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2 屬於 K，凸函數 (convex function) 是一個實函數 f:K\to\mathbb{R} 滿足下列性質：

對於任意 \mathbf{x}_1,\mathbf{x}_2\in K 且 0\le\lambda\le 1，

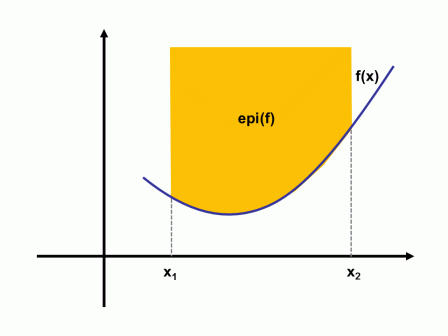
\displaystyle  f\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)\le \lambda f(\mathbf{x}_1)+(1-\lambda)f(\mathbf{x}_2)。

若定義式等號僅發生於 \lambda=0 和 \lambda=1，我們稱 f 是一個嚴格凸函數。

下圖為一凸函數範例



凸函數指上境圖（Epigraph (mathematics)）（圖像上方的點的集合）為凸集的一類函數。換言之，其圖像上，任意兩點連成的線段，皆位於圖像的上方。二階可導的一元函數f為凸，若且唯若其定義域為凸集，且函數的二階導數f''在整個定義域上非負。一元凸函數的熟知例子有二次x^2和指數函數e^x。



附圖為一上境圖，一個凸函數的上境圖 (在函數圖像上方的黃色區域) 為凸集。凸上境圖和凸函數是兩個等價的概念。另外，凸函數(convex function)的圖像形如開口向上的杯，凹函數(non-convex function)則形如開口向下的帽。

**(b)**

**Gurobi 影片中的Example轉換方法:**

Non-convex Q constraints:

Convert into bilinear constraints:

z11 := x12

z12 := x1 x2

z13 := x1 x3

z22 := x22

z22 := x22

z33 := x32

=>linear constraints:

從影片中的範例可以看出

並且根據以下規則

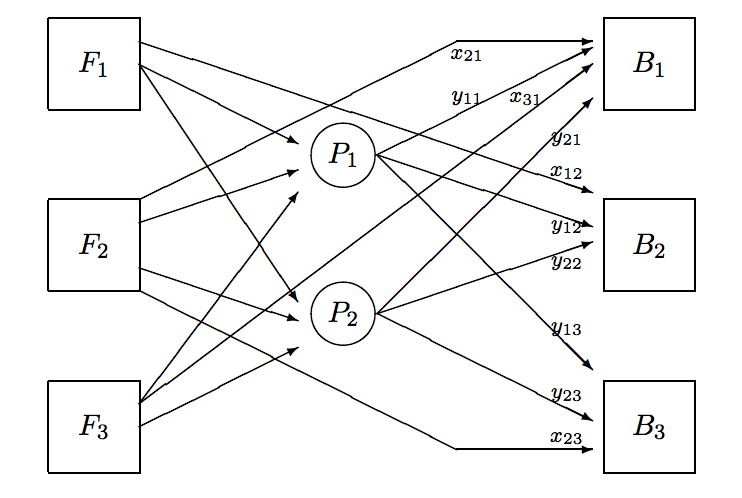
|  |  |
| --- | --- |
| if Q constraints is an equation |  |
| if Q constraints is inequality and |  |
| if Q constraints is inequality and |  |

**若以其中提到的pooling problem為例**

包括石化精煉、廢水處理和採礦等產業都常使用數值最佳化來解決池化問題。

它是產品具有不同規格的最小成本網絡流問題的概括。 解決池化問題的目標是在網路中找到滿足需求的最低成本流量。

下圖為一標準範例



雙線性限制(Bilinear)是一種非凸(non-convex)二次限制式的特殊形，如果我們想解決非凸二次限制，我們可以將這些約束轉換為雙線性約束，然後我們就可以使用 Gurobi 來解決問題。在這邊不討論求解，只討論Bilinear限制式。

This type of problems is typically solved using spatial **Branch and Bound** (老師上課提及的).

該演算法探索整個搜索空間，因此它為最佳目標值提供了一個全局有效的下界，並且如果有足夠的時間它會找到一個全局最優解，這邊我們以P-formulation 的方法來看。

以下為其中集合參數與決策變數

P-formulation (Concentration)

Sets and Indices

G=(V,E): Directed graph.

i,j∈V: Set of nodes.

(i,j)∈E⊂V×V: Set of edges.

N(i)+={j∈V∣(i,j)∈E}: Set of successor nodes receiving outflow from node i.

N(j)−={i∈V∣(i,j)∈E}: Set of predecessor nodes sending inflow to node i.

k∈Attrs: Set of attributes.

s∈Sources⊂V: Set of source nodes, i.e. N(s)−=∅.

t∈Targets⊂V: Set of target nodes, i.e. N(t)+=∅.

p∈Pools=V∖(Sources∪Targets): Set of pools.

Parameters

Costs∈R+: Cost of acquiring one unit of raw material at source node s.

Supplys∈R+: Maximum number of units of raw material available at source node s.

Contents,k∈R+: Content of attribute k in raw material at source node s.

Pricet∈R+: Price for selling one unit of final blend at target node t.

Demandt∈R+: Minimum number of units required of final blend at target node t.

Min\_tolt,k∈R+: Minimum tolerance for attribute k in final blend at target node t.

Max\_tolt,k∈R+: Maximum tolerance for attribute k in final blend at target node t.

Capp∈R+: Maximum Capacity to store intermediate blend at pool p.

UBi,j∈R+: Maximum flow from node i to node j.

Decision Variables

flowi,j∈[0,UBi,j]: Flow from node i to node j.

qualityp,k∈R+: Concentration of attribute k at pool p.

**一張含有 文字 的圖片

自動產生的描述**

上圖為目標式

**一張含有 文字 的圖片

自動產生的描述**

以上的限制式皆非Bilinear

一張含有 文字, 室外 的圖片

自動產生的描述

以上的限制式為Bilinear

該公式中Bilinear的數量與屬性的數量成比例。

另一種公式依賴fractions of flow而不是concentrations的決策變量，因此Bilinear項不再與屬性數量相關聯。

所以可以使用兩種類型的決策變量：

來自源頭 s 的池 p 總流入量的一部分。

在池 p 的總流出量的一部分，流向終端 t。

**(C)**

**詳細執行結果見ipynb檔**

**1.**

根據Gurobi的bilinear範例修改目標式跟限制式1

maximize: 2x^2 – 6xy + 3y^2

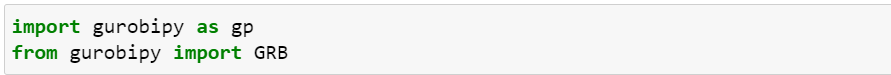
subject to:

x + 2\*y + z >= 12(linear constraint)

x\*y <= 8 (bilinear inequality)

x\*y + x\*z +y\*z == 3 (bilinear equality)

x, y, z non-negative (補充x是integral in second version)



一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

從影片及程式詳細執行結果可以發現Non-convex問題可以在Gurobi用bilinear的模型解，其模型會使用Cutting planes或Branching等方法來解決問題。(設定x為整數之執行結果)

**2.**

根據Gurobi的bilinear修改目標式跟限制式2

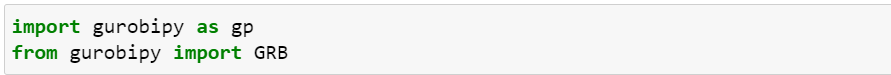
minimize : x+y

subject to:

2 \* x + y + 4 \* z <= 16(linear constraint)

x \* y + x \* z <= 17 (bilinear inequality)

x \* z + y \* z =18 (bilinear equality)

x, y, z non-negative (補充x是integral in second version) 

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

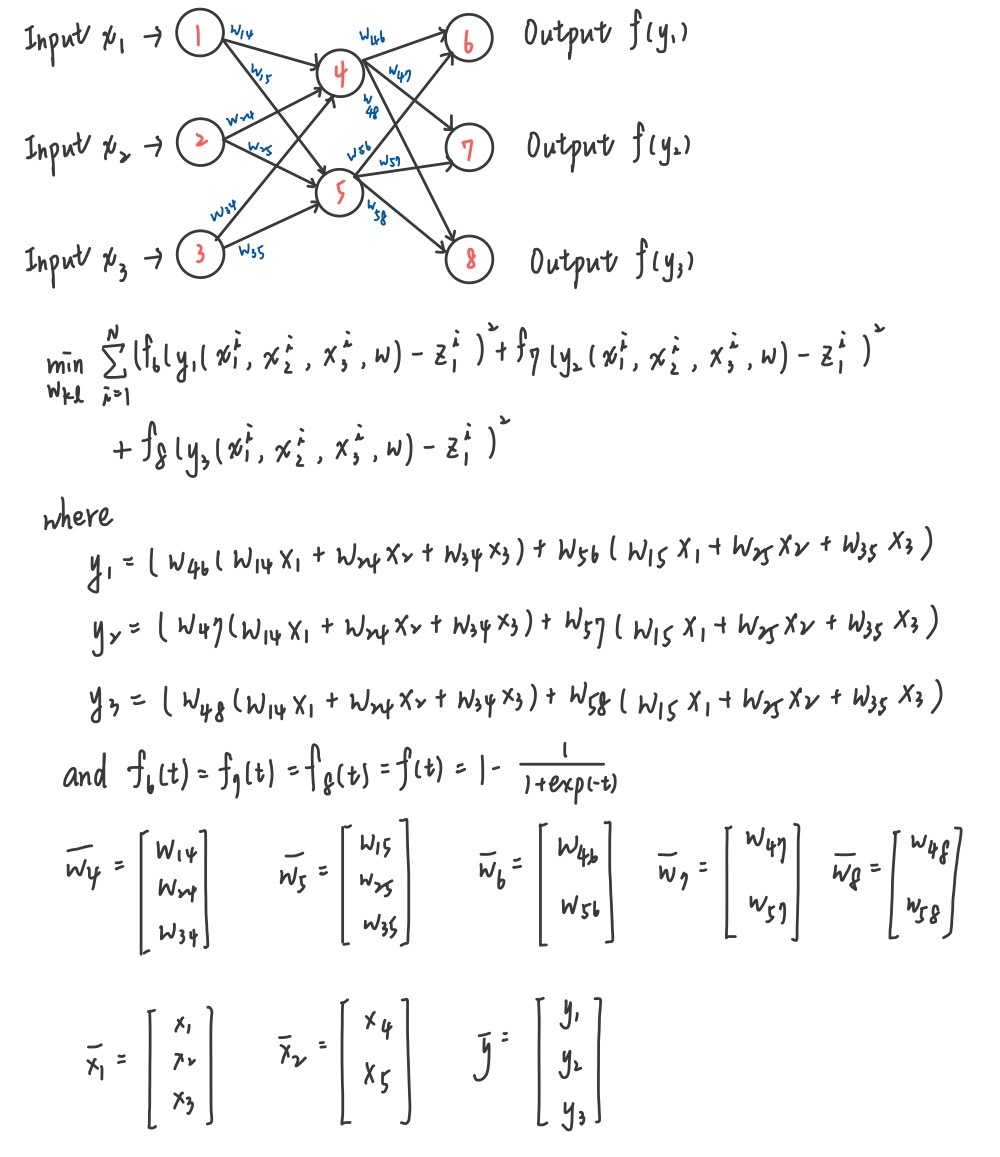
從影片及程式詳細執行結果可以發現Non-convex問題可以在Gurobi用bilinear的模型解，其模型會使用Cutting planes或Branching等方法來解決問題。(設定x為整數之執行結果)

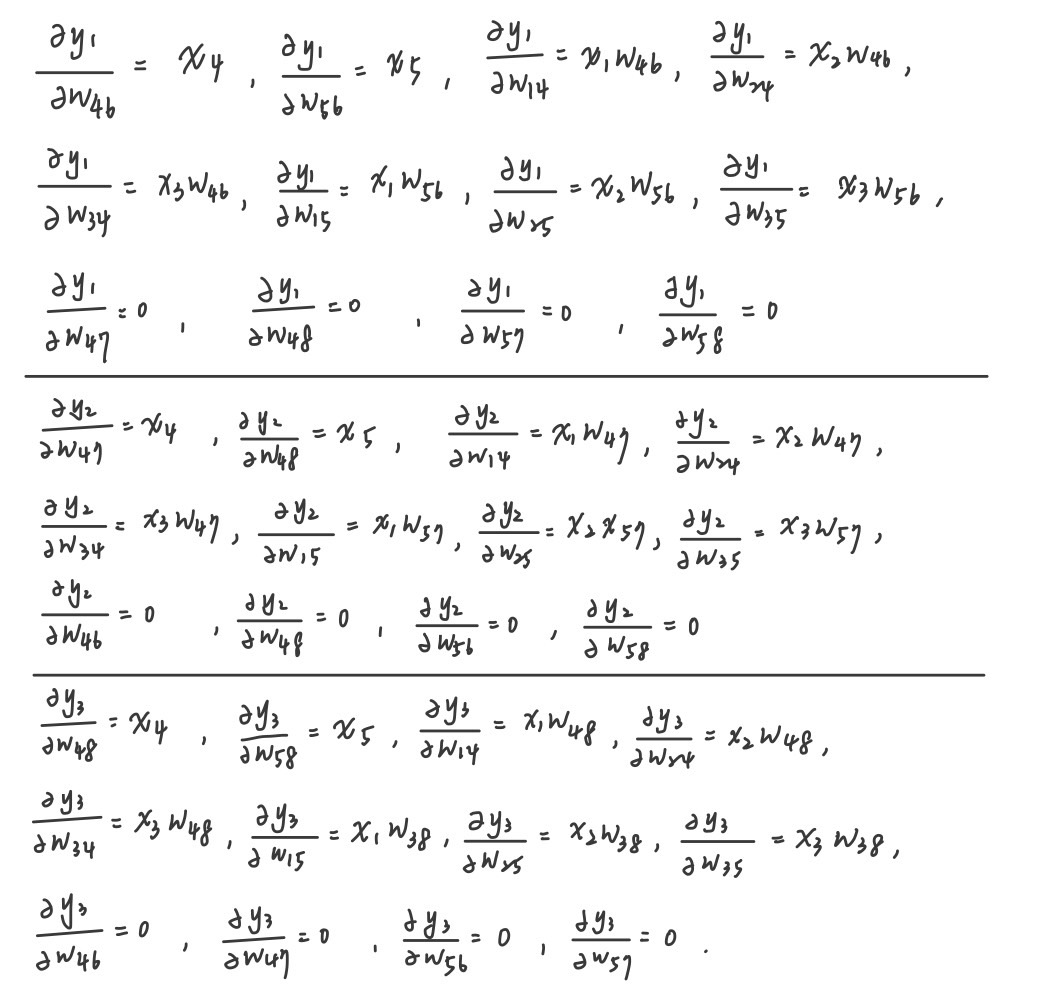
<https://colab.research.google.com/drive/1-oJ9PzUIUsYhTMEx53fG4AlYpCUTjL5u>

**第二題：**

（a）

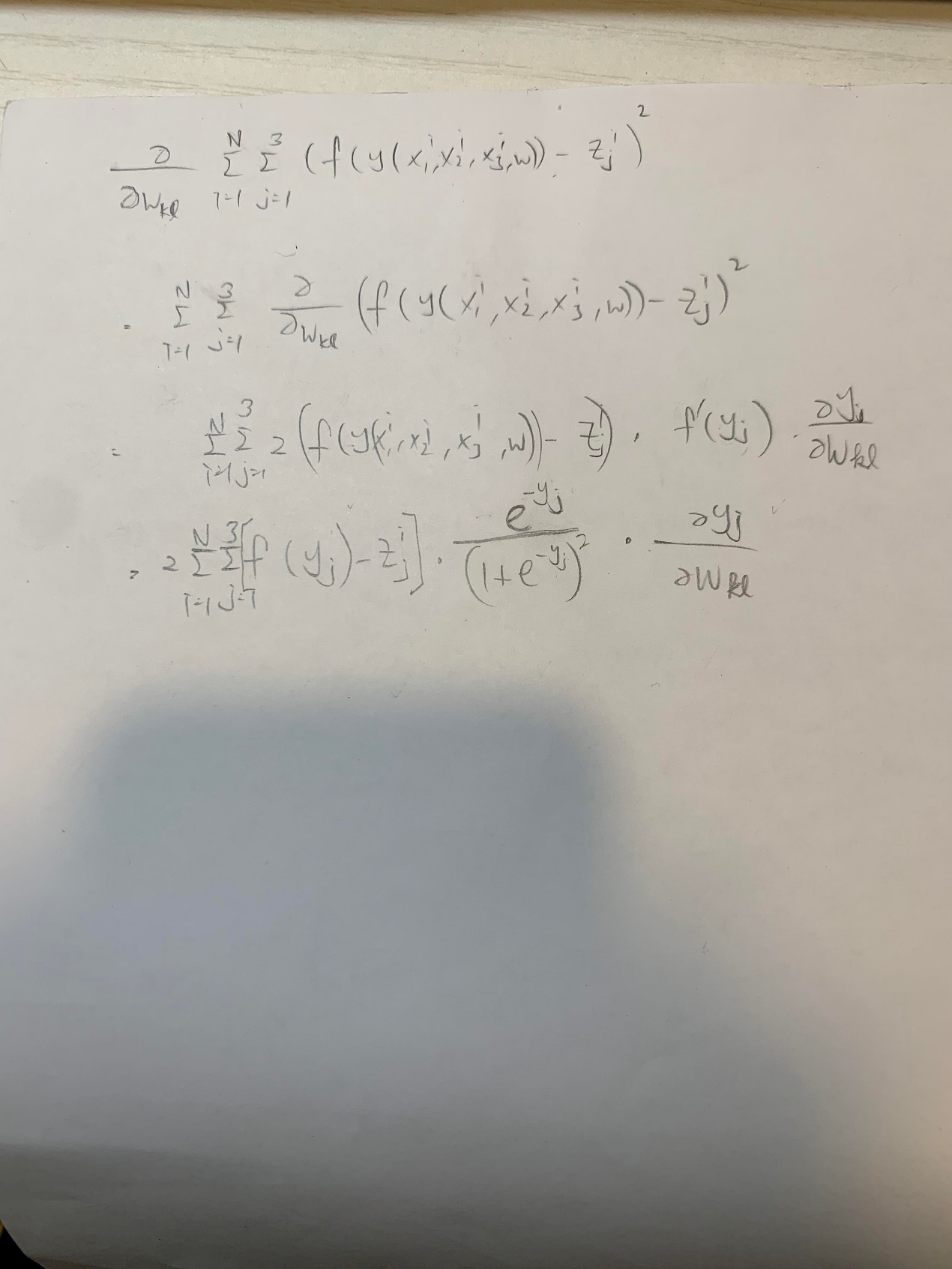
寫出function





（b）

將所求化簡可得到下列式子:



並以最後求得的數學式來算出所求之值。

在python上的執行如下:

1. 先設定所要input數值(x1,x2,x3)的數量及其所對應到之z值

一張含有 文字 的圖片

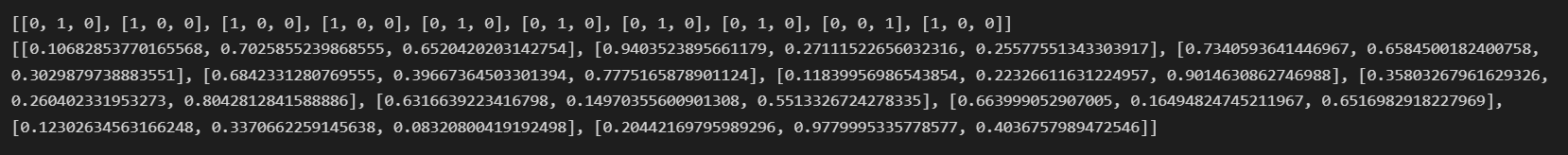
自動產生的描述

方法:

1. 在程式一開始先import random 和numpy的package

2. 設定亂數種子，將亂數的數值固定在以用14代入的亂數種子上

3. 先後設定input的資料數及z值的數目為10，其中z值的每筆資料裡又有3個數值

4. 用函數get\_dataPoint設定dataPointSet的數值並將(1,0,0)or(0,1,0)or(0,0,1)分配給z成為其數值。其中N為dataPointSet的數目，M為dataPointLabel單個的數目

所產生的資料值如上圖。

2.設定w4 w5 w6 w7 w8(arc)的數值，並將結果列印出來。

一張含有 文字 的圖片

自動產生的描述

方式:

用random.seed(14)亂數種子將將數值固定在14

用random.random()產生亂數。

設定一個function:init\_bar\_w4()來傳出bar\_w4 bar\_w5 bar\_w6 bar\_w7 bar\_w8的值

bar\_w4有三個數值:w14 w24 w34

bar\_w5有三個數值:w15 w25 w35

bar\_w6有兩個數值:w46 w56

bar\_w7有兩個數值:w47 w57

bar\_w8有兩個數值:w48 w58

將每一個的值用random.random()來給數值。

輸出結果如下圖。

一張含有 文字 的圖片

自動產生的描述

3.寫一個function get\_xy() 以得到x4 x5 y1 y2 y3 的數值

一張含有 文字 的圖片

自動產生的描述

方法:

1. 在function get\_xy()裡設定要輸入的參數值:x1,x2,x3,bar\_w4, bar\_w5, bar\_w6, bar\_w7, bar\_w8

2. 寫一個x的list來儲存x4、x5的資料值

並將X4 = x1\*w14 + x2\*w24 + x3\*w34 ，X5 = x1\*w15+ x2\*w25 + x3\*w35

用程式碼寫出

3. 寫一個y的list來儲存y1、y2、y3的資料值，並將

y1 = x4\*w46 + x5\*w56

y2 = x4\*w47+ x5\*w57

y3 = x4\*w48+ x5\*w58

用程式碼寫出

4. 最後return x y的值

5. 印出x y來查看數值

若x1=1,x2=2,x3=3

輸出結果如下圖。

一張含有 文字 的圖片

自動產生的描述

1. 輸入偏微分的值

一張含有 文字 的圖片

自動產生的描述

方法:

先創造一個function : get\_ydwkl(x, bar\_w6, bar\_w7, bar\_w8)，

再為y1 y2 y3的偏微分創造一個list:

ydwkl=[[0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0]]

再為y1 y2 y3的偏微分數值各自創造一個list:

🡪 dy1對應到y1的偏微分數值:

輸入y1對w46 w56 w14 w24 w34 w15 w25 w35 w47 w48 w57 w58的偏微分

🡪 dy2對應到y2的偏微分數值:

輸入y2對w46 w56 w14 w24 w34 w15 w25 w35 w47 w48 w57 w58的偏微分

🡪 dy3對應到y3的偏微分數值:

輸入y3對w46 w56 w14 w24 w34 w15 w25 w35 w47 w48 w57 w58的偏微分 一張含有 文字 的圖片

自動產生的描述

再寫出上面的程式印出每個偏微分對應到的數值。

4.用for迴圈將一開始推導出的公式程式化，並將產生的10個資料值代入公式，得到最後的答案。

方法:

1. 設定兩個function一個是將y值代入F的函數中，另一個是將y值代入F微分後的函數dF中

2. 呼叫function get\_dataoint(N, M) 並將值指定給dataPointSet 及dataPointLabel

3.呼叫init\_bar\_w4()並將值指定給bar\_w4 bar\_w5 bar\_w6 bar\_w7 bar\_w8

4.將result的初始值設為0

5. 寫一個for迴圈將10筆資料值代入input(x1 x2 x3)，並算出後面的y1 y2 y3值

6. 呼叫function get\_xy() 傳入的參數是上面設定好的dataPointSet，並將值指定給xy

7.呼叫get\_ydwkl()並將值指定給ydwkl

8.設定一個F的list 並用functionSig()將y值代入。

9.設定一個dF的list 並用dfunctionSig()將y值代入。

(此函數是functionSig()的微分結果)

10.寫一個for迴圈讓程式去跑j=0 1 2(y1 y2 y3)的值

11.再寫一個for迴圈讓程式去跑偏微分值

12.最後一行result += (F[j]-dataPointLabel[i][j])\*dF[j]\*val

是先將F()-z值再乘上微分後的F()，再乘上y的偏微分值。

並將加總後的結果指定給result

13.最後print()出result的值，即為所求。

<https://colab.research.google.com/drive/1yq2aBQJlk8M2qP7_kmnGWRwlIUzjuLvl>

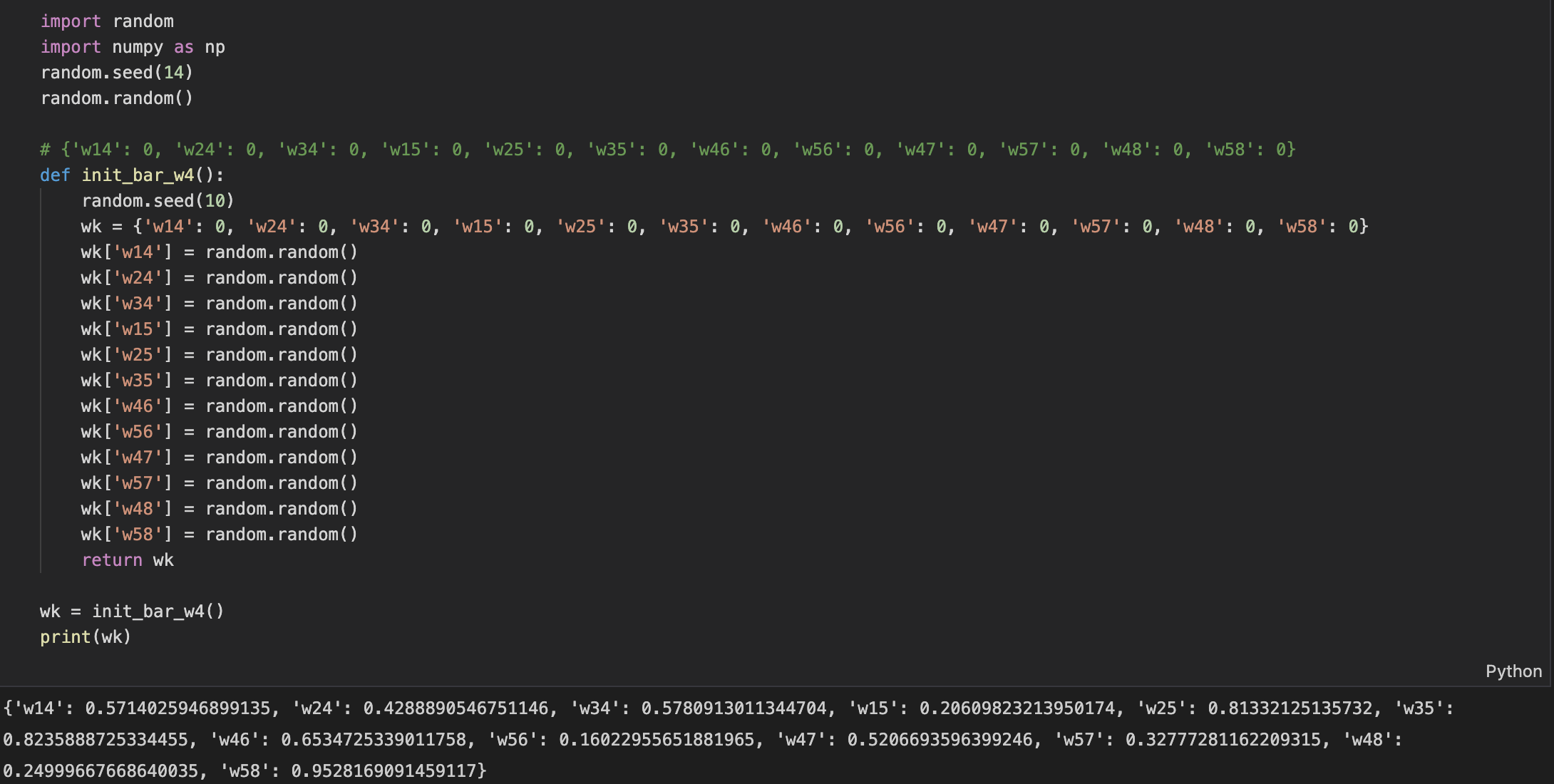
（c）

🡪設函數get\_dataPoint，印出所有的dataPointLabel、dataPointSet

一張含有 文字 的圖片

自動產生的描述

🡪並用dictionary儲存每個向量的值



let G = ( fA( y1(x1, x2, x3, w) ) - z1) )^2

let fA(t) = fA( y1(x1, x2, x3, w) )

G = (fA(t) - z1) ^ 2

dG/dw = dG/dt \* dt/dw

= 2 \* (fA(t) - z1) \*dy1/dw

= 2 \* (1/(1+np.exp(-y1)) - z1) \* dy1/dw

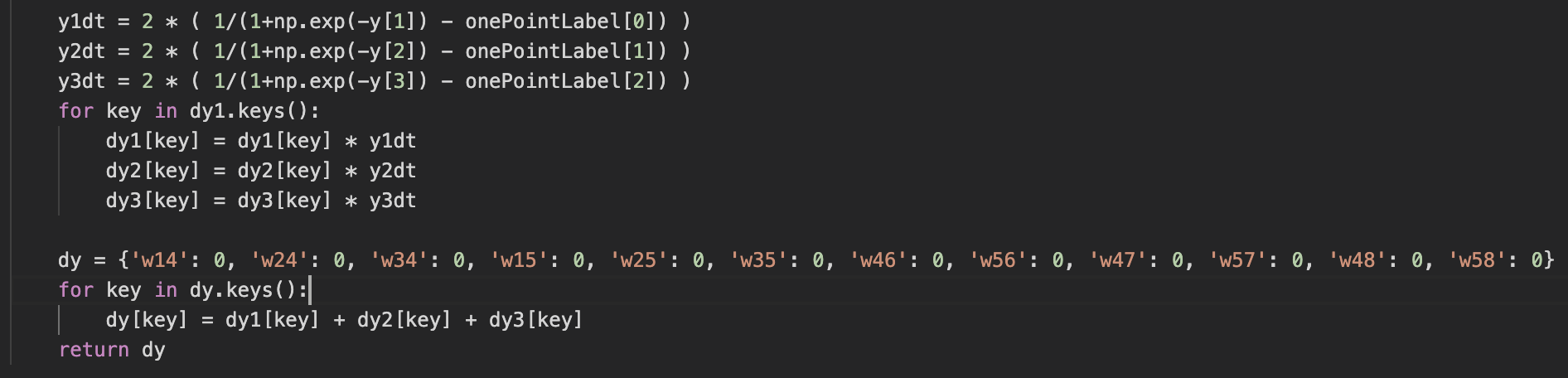
G = fA( y1(x1, x2, x3, w) ) or fB( y2(x1, x2, x3, w) ) or fA( y3(x1, x2, x3, w) )

fA(t) = fB(t) = fC(t) = 1/(1+np.exp(-y[1]))

🡪再把所有微分後的key值做加總

一張含有 文字 的圖片

自動產生的描述



🡪寫出函式nextWK，wk等於current point，dk等於gradient vector，新的wk(current point)會等於wk-dk

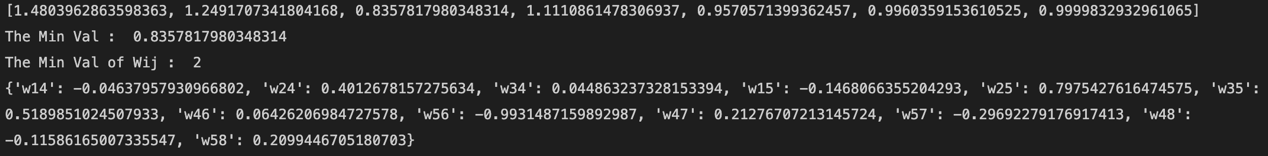


🡪最後利用以上各種data值，取出optimal value，(np.linalg.norm(list(dwk.values()))🡪這裡的list會將value值轉為矩陣)

一張含有 文字 的圖片

自動產生的描述

🡪得出結果



<https://colab.research.google.com/drive/1foWh_CsjKYmuLdQE7W9LYPmQoPU9mmeh>

**第三題：**

馬克維茨投資組合理論的基本假設為：(1)投資者是風險規避的，追求期望效用最大化；(2)投資者根據收益率的期望值與方差來選擇投資組合；(3)所有投資者處於同一單期投資期。以期望收益E來衡量證券收益，以收益的方差表示投資風險。資產組合的總收益用各個資產預期收益的加權平均值表示，組合資產的風險用收益的方差或標準差表示，則馬克維茨優化模型如下：

* n: 可供選擇的資產種類數量
* ri: 各種資產的預期收益率(i=1~n)
* σij: 資產ij之間的共變異數，也可寫為Cov(ri,rj)
* rp: 預期總收益率
* 決策變數 wi: 資產i在組合中的權重(0⩽wi⩽1)
* 投資風險為 D(rp)

**一張含有 文字 的圖片

自動產生的描述**

Eq.(1) minimizes the total variance (risk) associated with the portfolio.

Eq.(2) ensures that the portfolio has an expected return of R.

Eq.(3) ensures that the proportions add to one.

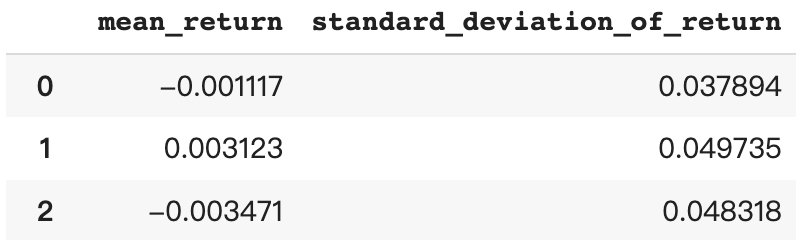
補充：correlation = σij ÷ (σ𝑖∗σ𝑗)

* 用矩陣表示的話，Q為共變異數矩陣



****

利用pandas套件讀取port5.txt，df1代表第1~225行的data frame，sep=’ ‘代表利用空白區分資料，所以第1行資料變為”NaN”、”-.001117”、”0.037894”，再用usecols取後面兩欄的資料，最後將欄位命名為「期望報酬率」與「標準差」，結果如下：



同樣的方法取出第226~25425行的資料，df2即為存放資產i與j之間相關係數的data frame（注意不是共變異數，所以後面要將相關係數乘上標準差）。

一張含有 桌 的圖片

自動產生的描述

**一張含有 文字 的圖片

自動產生的描述**

設定n為225種，從port5.txt中給的報酬率來看，rp可以設置為0.001以下。

一張含有 文字 的圖片

自動產生的描述

先將ri設為長度為n的零矩陣（numpy），再用迴圈逐一讀入報酬率。

一張含有 文字 的圖片

自動產生的描述

先將σij設為n\*n的二維零矩陣，這邊要先注意在port5.txt已經出現過的i與j的相關係數不會再重複出現（例如i=1、j=2的相關係數出現過，那i=2、j=1的就不會再出現）。所以用迴圈讀取相關係數後（再乘上標準差），就會得到共變異數上三角矩陣。因為下三角矩陣其實就是對稱的上三角，最後將這個σij矩陣加上自身轉置後減掉對角線元素，即為共變異數矩陣

(1) 相關係數上三角矩陣 一張含有 桌 的圖片

自動產生的描述

(2) 共變異數上三角矩陣 一張含有 文字 的圖片

自動產生的描述

(3) 共變異數矩陣 一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

加入n個權重wi，用numpy轉為矩陣形式

一張含有 文字 的圖片

自動產生的描述

加入限制式，各資產報酬率ri乘上權重wi後總和為預期總報酬率rp

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

目標式乘上1/2只是為了方便運算，對結果不影響。

<https://colab.research.google.com/drive/16J_5dC8BQs7yDK3_7U4hm7w0FPEbXIkS>