

basic part

我這次用的演算法是式 **gradient descent**，我以下會講解每個 **function** 的功能

1. SplitData:

我把 **data** 分成三個部分，第一筆有 159 筆資料當作 **train data**，第二筆有 30 筆資料當作 **validation data**，以便於測試我 **model** 的準確度，第三筆有 20 筆資料，預測 10/15~11/11 的資料。

2. PreprocessData

分別輸入分為兩個部分，MTK 的 **data** 當作 **X**，TSMC 的 **data** 當作 **Y(ground truth)**。因為我的 $f(x)=w_4x^4 + w_3x^3 + w_2x^2 + w_1x + w_0$ ，所以我的 **X** 會變成陣列，變成 $[x^4, x^3, x^2, x, 1]$ 。

3. Regression

核心概念就是 $w=w-\text{learning_rate}*g$ ，**w** 的直要不斷被更新，我的 **w** 的值一開始是設為 $[0,0,0,0,0]$ ，而 **g** 的值會在 **CountLoss** 中被計算出來。而每個係數都有不同 **learning_rate**，經過不斷的測試，我將 **learning_rate** 設為 $[10^{-29}, 10^{-21}, 10^{-15}, 10^{-9}, 10^{-3}]$ ，而 **iteration** 設為 20000 次。當 **iteration** 跑到 20000 次的時候，**w** 的值基本上已經收斂了。

4. CountLoss

這個 **function** 的作用就是計算 **g**，用到的公式就是講義上的公式，如下圖

• For $t=1, \dots$ until convergence:

• Predict for each example \mathbf{x}^i using **w**: $\hat{y}^i = \sum_{j=0}^k w_j \phi_j(\mathbf{x}^i)$

• Compute gradient of loss: $\frac{\partial}{\partial w_j} J(\mathbf{w}) = 2 \sum_i (y^i - \hat{y}^i) \phi_j(\mathbf{x}^i)$

• This is a vector **g**

5. MakePrediction

最後一步就是拿計算好的 **w**，去乘上 **test data**。我最後的 **w** 為 $[-2.34478473\text{e-}11, -1.29854370\text{e-}07, -2.06041151\text{e-}04, 9.42738624\text{e-}01, 2.15721864\text{e+}01]$

以上就是我每個 **function** 的作用，以下會再補充說明幾點:

1. 我另外有寫了一個 **evaluate function** 以測試我的 **model** 的準確度，用的方法是 ppt 上寫的算法(如下圖)，我在 **train model** 時，也是看 **mape** 的值去判斷當前 **model** 的好壞。做測試的時候，我拿 **validation data** 去跑 **model**，得到的值是 2.59%。

$$\text{MAPE} = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

bonus:

我這次用的演算法是式 **gradient descent** 與 **Autoregressive(AR)**，除了 MTK 之外，我還有用 UMC(聯電)今年的 data，因為他的產業發展跟台積電蠻像的。有些部分跟 basic 很像，所以以下會針對不一樣的地方做講解。

做法主要分兩個部分，第一個是用 AR 的方式，分別預測 MTK 跟 UMC 在 10/15~11/11 這 20 天內的價錢。第二個部分則是運用 basic 的方法，用 1/1 到 10/14 的 data，找出 MTK 跟 UMC 與 TSMC 的相關性(w)，再將這個 w 套用到之前 MTK 跟 UMC 預測出來的資料，由此預測 10/15~11/11 TSMC 的 data。以下會分成這兩個部分做講解：

第一部分:

我以 MTK 做講解(UMC 的做法是一樣的)，我的 train data 是取 MTK 前四天的 data 去預測今天的 data，而 training 的方法一樣是用 **gradient descent**。

$f(x)=w_4x_4 + w_3x_3 + w_2x_2 + w_1x_1 + w_0$ 。我的 w 的值一開始是設為 [-1,1,-1,1,30]([前四天,前三天,前二天,前一天,常數])，iteration 設為 20000 次，我會這樣設是因為經過不斷的測試，這樣設會有比較好的表現，而 g 的值會在 CountLoss 中被計算出來。CountLoss 的作法跟 basic 是相同的。

另外，因為 UMC 的股價比較低(約 60 塊)，所以我在處理資料的時候，會先呈上 10 倍，在進行 train 跟 predict 的動作。還有因為 MTK 跟 UMC 的 data 在價錢上有差異，所以我用不同的 learning_rate 去 train 這兩家公司，分別是 10^{-7} 跟 10^{-5} 。

第二部分:

接下來的部分就跟 basic 很像了，用三家公司 1/1 到 10/14 的 data 去做訓練，MTK 跟 UMC 的 data 當作 x_1 跟 x_2 ，TSMC 的 data 當作 Y， $f(x)=w_2x_2 + w_1x_1 + w_0$ 。w 的值一開始是設為 [0,0,0]，learning_rate 都是 10^{-7} ，iteration 為 20000。

最後再將第一部分預測出來的資料，乘上第二部分算出來的 w，就可以得到 10/15~11/11 TSMC 的股價。