**Sample Code & 作業內容**

Coding練習日，請結合前面的知識與程式碼，比較不同的 optimizer 與 learning rate 組合對訓練的結果與影響，作業請提交Day080\_HW.ipynb

**[今日百日馬拉松作業 : 注意事項]**

1. 今日作業記憶體需求較大, 請同學執行時記得 shutdown 其他執行中的 .ipynb 檔

2. 提醒同學們今日執行會比較久(幾分鐘到數小時不等)

[檢視範例](https://ai100-2.cupoy.com/samplecodelist/D80)

**參考資料**

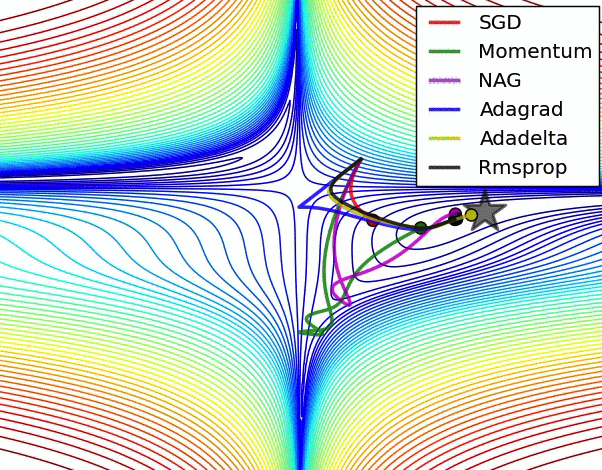
CSDN - 常見優化器方法總結：

* SGD (mini-batch)
  + 在單步更新與全局更新的折衷辦法，通常搭配 momentum 穩定收斂方向與結果。
  + 收斂速度較慢。
* RMSprop
  + 學習率的調整是根據過去梯度的狀況調整，收斂速度快又不易會出現 learning rate 快速下降的狀況。
* Adam
  + 同樣是可以根據過去的梯度自行調整 learning rate，但校正方式考量一、二階矩陣，使其更加平穩。

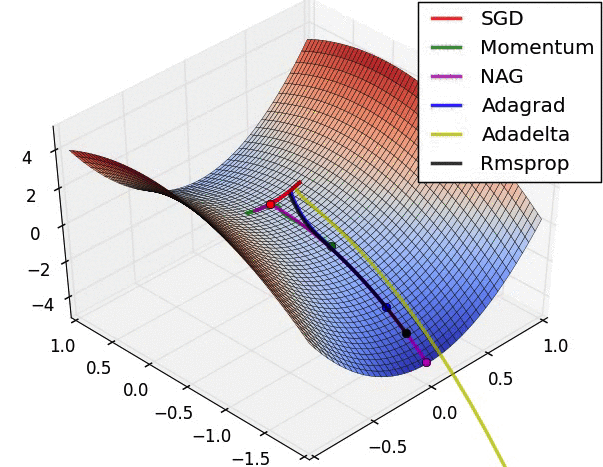
在實作過程中，建議先使用 Adam 驗證，若要做最終的優化，則再改用 SGD 找到最佳參數。

參考連結：

* [**http://cs231n.github.io/assets/nn3/opt2.gif**](http://cs231n.github.io/assets/nn3/opt2.gif)



* [**http://cs231n.github.io/assets/nn3/opt1.gif**](http://cs231n.github.io/assets/nn3/opt1.gif)



* [**http://ruder.io/optimizing-gradient-descent/**](http://ruder.io/optimizing-gradient-descent/)
* [**CSDN-優化器方法總結**](https://blog.csdn.net/u010089444/article/details/76725843)