**Sample Code & 作業內容**

請參考範例程式碼Day082\_Dropout.ipynb，完成以下作業：

作業1：請比較使用不同層數以及不同 Dropout rate 對訓練的效果

作業2：將 optimizer 改成使用 Adam 並加上適當的 dropout rate 檢視結果

請提交Day082\_HW.ipynb

[檢視範例](https://ai100-2.cupoy.com/samplecodelist/D82)

**參考資料**

**CSDN - 對 Dropout 的詮釋**

* 組合派 (Ensemble)

        overfitting → 參數量過多 → 找到一個比較瘦的網路可能可以降低 overfitting 的程度 → 手動或使用 Grid-

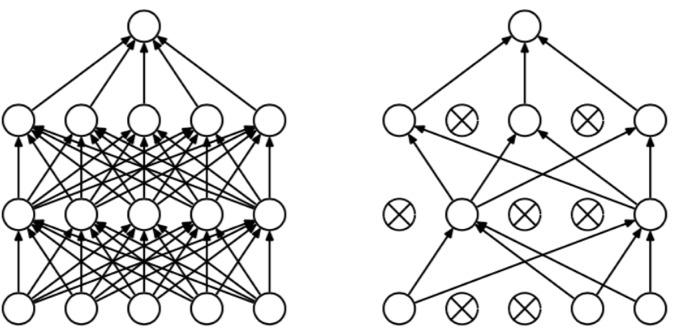
        Search? 太慢 → 使用 Dropout 使學習時隨機打斷一些連結 → 可以解決費時的問題，把重要的連結權重增

        加；不重要的減輕 → 重新組合起來後 (Inference phase) 自然會是一個以權重定義後的瘦網路。

* 動機論 (有豬隊友)

        overfitting →  有人 (參數/連結) 濫竽充數 → 隨機將某些人 (神經元) 分成數組 → 各組學會懂得自立自強

        → 重新組合後 (Inference phase) 會比原本更強大。



參考連結：

* [**理解 Dropout – CSDN**](https://blog.csdn.net/stdcoutzyx/article/details/49022443)
* [**Dropout in Deep Learning**](https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5)