Machine Learning HW14 Report

學號：B06901045 系級：電機三 姓名：曹林熹

1. (2%) 請以中文說明一下 lifelong learning 的中心概念是什麼？

ANS：

Lifelong learning 的中心概念就是讓機器可以在有新的訓練 task 時，使用同一個神經網路將此 task 做好，同時不會遺忘先前所學好的 task 參數，類似一個模型可以學習多種技能的概念 (Knowledge Retention but NOT Intransigence)。此作法出現的背景可以概括以下兩個方面：

1. 隨著科技的進步，各種資料呈爆炸式增長，如何有效率地在資料獲得時馬上學好新的任務是關鍵。

2. 傳統機器學習演算法對大資料環境下的應用問題很多都不再適用，因為傳統的機器學習演算法多數關注小樣本的分類工作等，對大資料環境缺乏學習與適應的能力。

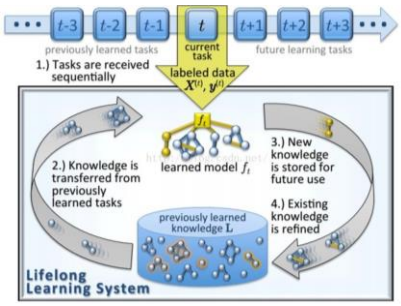
以下四點是基本要素，而 Lifelong learning 的學習框架，可以用下圖來說明：

1. 維護可增長的資料庫：長期儲存知識且對儲存資料有效的檢索和再現能力

2. 按照一定順序學習：每個資料訓練是有條不紊的

3. 多個任務：可以對多個不同任務資料進行測試

4. 知識的正向遷移：選擇對新模型有用的知識進行遷移，幫助新模型的學習



2. (2%) 列出 EWC, MAS 的作法是什麼？根據你的理解，說明一下大概的流程該怎麼做 (不要貼 code)。

ANS：

在做 multi-task 時，其實可以將所有的資料重新合併成一本新的 dataset 再重新 training，但是如果使用這種方法，非常耗時間與空間。因此我們希望機器學習新的參數時，只使用當前訓練的 dataset，且不要大幅更動到以前好的參數，只更動以前不重要的參數 (avoid Catastrophic Forgetting)。以下介紹 EWC 與 MAS 算法實現此目標。

1. EWC

主要做法是將 loss function 改寫，新的 loss function 是 L_B，他是當前訓練任務的 loss function L(θ) 加上一個 regularization term。此 regularization term 與先前的訓練任務參數有關，F_i 代表先前的訓練參數 $\theta^*_{A,i}$ 有多重要，F_i 愈大代表參數愈重要，那我們就會讓當前學習的新參數 θ_i 與 $\theta^*_{A,i}$ 差距小一點，才不會讓 loss function 太大。在 summation 前面有個係數 $\frac{\lambda}{2}$ 需要人工設定，最後就是 training 時使 L_B 愈小愈好。

$$\mathcal{L}_B = \mathcal{L}(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{A,i})^2$$

而 F_i 的求得方法，助教很貼心地給出了下列公式。$p(y_n|x_n,\theta^*_A)$ 代表 posterior probability (後驗機率) 的概念，x_n 則代表之前 task 的 data，θ^*_A 則代表訓練完 task A 存下來的模型參數，y_n 則是對應的 x_n label。最後，針對這個後驗機率取 log 後再取 gradient 的平方即是我們的 F。

$$F = [\nabla \log(p(y_n|x_n,\theta^*_A) \nabla \log(p(y_n|x_n,\theta^*_A)^T]$$

2. MAS

MAS paper 介紹：https://arxiv.org/abs/1711.09601

主要做法也是將 loss function 改寫，與 EWS 方法相似，不同的地方為 F_i 係數

改成 Ω_i 。

$$\mathcal{L}_B = \mathcal{L}(\theta) + \sum_i \frac{\lambda}{2}\Omega_i(\theta_i - \theta^*_{A,i})^2$$

而 Ω_i 由下列的式子給出，x_k 則代表之前 task 的 sample data，其中 l_2 為 12 norm，代表歐幾里得距離，在網路資源上有給出了定義的公式。我們將最後一層取 12 norm 的平方，再根據此值做 gradient 且最後加絕對值。

$$\Omega_i = ||\frac{\partial \ell_2^2(M(x_k;\theta))}{\partial \theta_i}||$$

$$\|v\|_2 := \sqrt{\sum_{i=1}^{n} |v_i|^2} = \sqrt{v^T v}, \text{ 2-norm}$$

3. (1%) EWC 和 MAS 所需要的資料最大的差異是什麼？

ANS：

EWC 是針對 supervised 的資料，MAS 則是 unsupervised 的，因此 MAS 可以使

用沒有 label 的資料。此外，在有關於 MAS 的論文中，作者提到使用 MAS 計算重要的 weight 時，是針對 the sensitivity of the output function 而不是 loss，可以減少 gradient 逼近 0 而跑進 local minimum 的機率。

Method	Type	Constant Memory	Problem agnostic	On Pre-trained	Unlabeled data	Adaptive
LwF [17]	data	✓	X	✓	n/a	X
EBLL [28]	data	X	X	X	n/a	X
EWC [12]	model	✓	✓	✓	X	X
IMM [16]	model	X	✓	✓	X	X
SI [39]	model	✓	✓	X	X	X
MAS (our)	model	✓	✓	✓	✓	✓

Table 1: LLL desired characteristics and the compliance of methods, that treat forgetting without storing the data, to these characteristics.

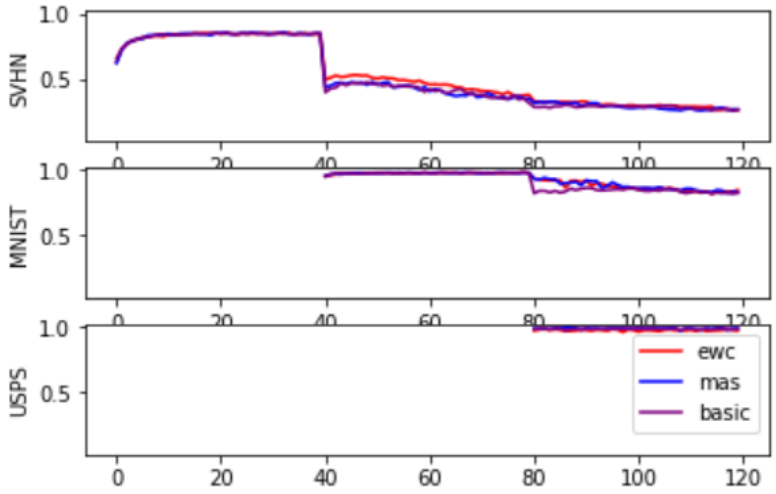
4. (5%) 秀出 part1 及 part2 最後結果比較圖，並分析一下結果，以及你跑的實驗中有什麼發現。

(EWC, MAS, baseline 比較圖 (2%) 與 EWC、MAS、SCP (或是你自己實做的演算法)、baseline 比較圖 (3%))

ANS：

1. EWC, MAS, baseline 比較圖

下圖給出了我們參數都沒調的圖片，預設值 λ 皆為 0。可以看到沒有調參數的狀況下，EWC、MAS、BASIC 在三個測試資料集的表現都是差不多慘的。



助教很貼心地給了我們 lambda 超參數的建議：ewc：80~400、mas：0.1~10。

而稍微調整超參數，經過幾次與同學合作的 trial and error 後，實驗結果發現在 λ_EWC = 400, λ_MAS = 0.1 有著較好的表現，結果如下圖。

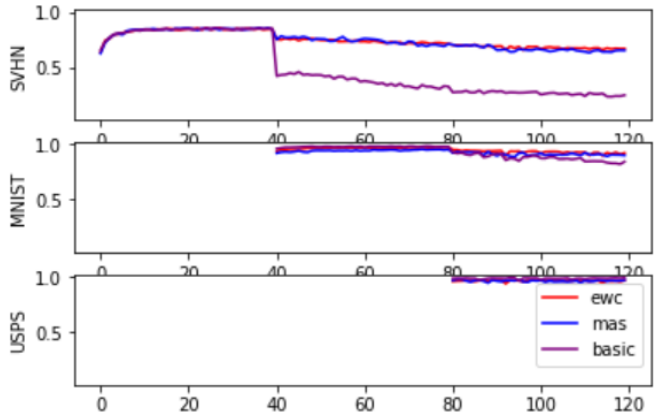
首先我們分析 SVHN，這是一個彩色的數字圖片資料集。只訓練 SVHN 去看準確率時，準確率大約為 7~8 成左右。當我們訓練完 MINST 與 USPS 再回去看 SVHN 的表現，可以發現 EWC 與 MAS 的表現比一開始稍差一點，但是表現算是穩定而沒有太多的遺忘，可見經過這兩個方法是有效的成果。

接著我們看到 MINST，訓練完 SVHN 再訓練 MINST 後可以有逼近 100% 的準確率，可見我們的模型並沒有發生 Intransigence。當我們訓練完 USPS 再回去看

MINST 的表現，也比起初沒有調整合適的超參數好。

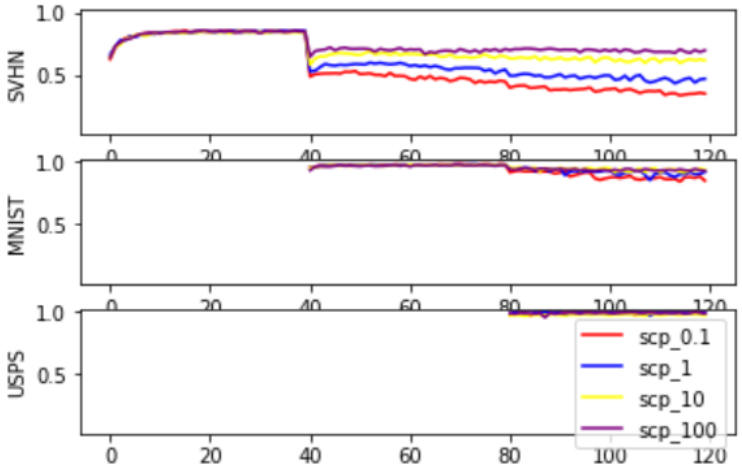
最後看到 USPS，三個方法都有不錯的成績，與前面相同。

總結來說，我認為使用 EWC 與 MAS 可以大幅增進 Knowledge Retention，不會遺忘先前的 task。



2. EWC、MAS、SCP、baseline 比較圖

在這裡我多使用了 SCP 方法，L 採用 100。由於此方法一樣要調超參數，因此我先做了四種不同的超參數值，分別為 0.1、1、10、100。可以看到在 SVHN 上，當我們的超參數愈大，表現效果愈好，而在 MINST 與 USPS 則表現差不多。



最後，我挑出一組超參數拿去與 EWC、MAS、BASIC 一同做比較，此時超參數分別為：λ_EWC = 400、λ_MAS = 0.1、λ_BASIC = 0、λ_SCP = 100，可以看到其實 SCP、EWC、MAS 表現上是差不多的。在訓練時間的部分，我訓練的 SCP 會耗時比 EWC、MAS、BASIC 都還久，而且使用不同的 SCP 超參數值訓練時間也不太相同(超參數愈大訓練愈久)。總結而言，此次作業訓練時間真的久，因此在相同的表現結果下，我認為使用 EWC、MAS 會比較有效率。