1. 解釋程式碼

* select\_user()：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

若輸入的num\_users大於self.users，就將全部的self.users回傳。反之，就隨機挑選num\_users個users回傳。

* aggregate\_parameters()：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

首先將原本server的model parameters清0，方便之後直接拿users的parameters取代。再來就將每個users的parameters乘上「users所使用的train\_samples在全部samples當中所佔的比例」，並加到server的model parameters當中，完成aggregate。

* set\_parameters():

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

根據beta，將global parameters與local parameters做結合。結合的方式為：beta \* global model parameters + (1 - beta) \* local model parameters。

1. 探討問題
2. data distribution
3. alpha=0.1

一張含有 文字, 螢幕擷取畫面, 字型, 黑與白 的圖片

自動產生的描述

一張含有 文字, 字型, 螢幕擷取畫面, 黑色 的圖片

自動產生的描述

當alpha為0.1時，資料在每個user上面分佈的非常不平均，所以的users都只有特定label的資料，導致每個local model都無法訓練的非常全面，無法平衡地學習到全部的features，因此也導致最後的global accuracy只有0.3左右。

1. alpha=50

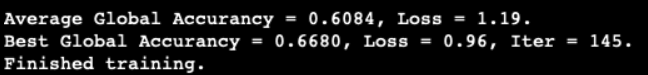
一張含有 文字, 螢幕擷取畫面, 黑與白, 字型 的圖片

自動產生的描述



當alpha為50時，資料在每個user上面分佈的相對平均，所以的users都擁有全部label的資料，因此每個local model可以平衡地學習到全部的features，最後的global accuracy也達到0.78左右。

1. Number of users in a round
2. num\_users=2



1. num\_users=10

一張含有 文字, 字型, 螢幕擷取畫面, 黑色 的圖片

自動產生的描述

資料在每個user上面分佈的相對平均的情況下，可以看到每一輪更新10個users對global accuracy有著更想顯著的幫助。這是因為每個users都可能擁有獨特的數據，因此綜合每個users的參數更新都可以讓global model在整體上學習的更好。至於收斂的速度，我的實測結果為這兩種訓練方式都在120~130個epochs收斂。

1. 最終acc的輸出

一張含有 文字, 字型, 螢幕擷取畫面, 黑色 的圖片

自動產生的描述

1. 此次作業中學到的重點

在trace code的過程當中，使我對federated learning有更深刻的了解。

1. 我了解到server如何選擇要更新的user model，也學習到server與users如何根據收到的參數，更新自身的model。
2. 並了解每個users所擁有的資料分布，對於最後的global accuracy會有很大的影響。server需要適應不同的數據分佈，並綜合考慮各個使用者的貢獻，以獲得更好的全域準確度。
3. 每一輪更新的users數量，對global accuracy也會有影響。如果每輪僅選擇少數users與模型更新，可能會造成部分users的資料未被充分利用，導致模型在表現不佳。