## 1. Logistic Regression

這部分幾乎和上課教的一樣。輸出是 linear regression 再取 logistic 函式 $\sigma$：

$$f_{w,b}(\vec{x}_n) = \sigma(\sum_{m=1}^{M} w_m x_{n,m} + b)$$
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Loss function 是把每筆資料 $\vec{x}_n$ 對 label $y_n$ 取 cross-entropy，再取平均：

$$L(f) = \frac{1}{N}\sum_{n=1}^{N} C(f(\vec{x}_n), y_n)$$
$$C(f(\vec{x}_n), y_n) = -[y_n \ln f(\vec{x}_n) + (1-y_n)\ln(1-f(\vec{x}_n))]$$

和講義不同在於取了平均，目的僅為觀察平均單筆資料誤差，不影響收斂演算法。Loss function 對 $w_m$ 偏微分，再用 gradient descent 更新 $w_m$：

$$w_m \leftarrow w_m + \frac{\eta}{N}\sum_{n=1}^{N}(y_n - f(\vec{x}_n))x_{n,m}$$

其中 learning rate $\eta$ 會用 AdaGrad 方法更新；我沒用 mini-batch (全部資料一起算一次更新)；更新 bias $b$ 方法類似。實作細節可看我程式 (位置 `poop/logisregress.py`)。

## 2. 其它方法: Random Forest

除 logistic regression (LR) 我還試了 neural network (NN)、decision tree (DT)、random forest (RF) 三種方法。交叉驗證 (稍後說明) 發現 RF 效果最好，故選它當我第二個方法。RF 基本上就是建一堆 decision trees，每棵 DT 建法是：

1. 假設有 $N$ 筆資料，則隨機選 $N$ 筆 samples，但可重複，再把重複的踢除。英文是 random samples with replacement。
2. 假設有 M 個 features，則隨機選 $m$ 個出來 ($m \leq M$)。
3. 用挑出的 samples 和 features 建一棵 DT。
4. 重複上述步驟，建立夠多的 DTs。

這樣 forest 就建好了。要驗證或測試時：

1. 把某筆 data 餵給每棵 DT，但要用每棵 tree 指定的 $m$ 個 features。
2. 每棵 DT 會預測出一個分類，最後用「多數決」看哪個分類票數最多。

我 DT 是用 Gini index 和 binary tree，對每個 feature 會用 Gini 值決定門檻，看哪個 feature 的 Gini 值最低，就用它和對應的門檻值，將資料切左右兩邊，沒有 pruning。DT 其它細節可參考我程式實作。DT 和 RF 程式位置在 `poop/dectree.py` 和 `poop/randforest.py`。

### 3. 用交叉驗證選模型和實驗結果

我用前面提的四種方法 (LR、NN、DT、RF)，每個都試不同參數和 features，然後用 N-fold 交叉驗證 (cross-validation) 挑選模型。交叉驗證演算法如下：

1. 把資料隨機 shuffle。
2. 取前 $K$ 筆為 validation set，$K$ 接近 test set 大小 (約 600 筆)。剩下資料當 training set。
3. 用選擇的模型 (方法 + 參數 + features) 和 training set 做訓練，再用 validation set 計算誤差或準確度。
4. 回到 1. 再訓練再驗證，重複 $N$ 回合。
5. 把 N 次的誤差或準度取平均，當作這模型的結果。
6. 選擇平均誤差最小的模型，我們相信它在 test set 表現也會最好。

這次作業使用 random forest 效果超級無敵顯著！冰山一角的實驗結果如下，它們都有把第 $55^{th}$ 和 $56^{th}$ 的特徵取 log 當新 features。

| 方法與參數 | Training 時間 | 驗證準度 |
|---|---|---|
| LogisticRegression(lrate=10.0, num_iters=500000) | 5 分鍾內 | 92% ~ 93% |
| RandomForest(num_features=10, num_trees=60) | 1 分鐘內 | 91% ~ 92% |
| RandomForest(num_features=30, num_trees=60) | 2 分鐘內 | 94% ~ 95% |
| RandomForest(num_features=30, num_trees=200) | 5 分鐘內 | 95% ~ 96% |

NN 在 training 準度和 LR 差不多，但 train 很久所以放棄；用 DT 容易 overfitting，也就是 training 準度逼近 100%，但 validation 準度只剩約 90%；然而 RF 設定 features 數 30，超過 50 棵樹，在 validation 準度穩定超過 94%！

最後再對 RF 調整參數。設定 features 數目在 25 ~ 45 效果差不多最好；使用的 trees 數量，可說是越多越好。當 tree 超過 150 棵，準度幾乎超過 95%。使用 200 棵樹，有時還會超過 96%！