

ML HW3 Report - Backpagation

Author

- 108062213 顏浩昀

Basic Part

1. Forward Pass

1. 首先在initialize_parameters_deep的地方將每一個layer的w和b設定好（在這裡透過固定random_seed的rand與0設定）
2. 接著透過linear_forword，將 $\text{dot}(W,A)+b$ 得到Z
3. 將Z放入Activation Function

這邊的Activation Function有sigmoid與relu兩種

1. sigmoid fn: 因為 $\exp(\text{num})$ 如果給予一個太大的num會有overflow的狀況，因此特別分為 $Z \geq 0$ 與 $Z < 0$ 的狀況。
2. relu fn: $\text{maximum}(0, Z)$
4. 最後就是將各層layer一層一層接起來(除了最後一層用sigmoid fn，前面n-1層都是使用relu fn)

2. Cost Function

1. Binary cross-entropy loss

用來計算2 classes的cross entropy

```
1 cost = np.sum( np.sum( (Y*np.log(AL)), axis=0, keepdims=True),  
2                 axis=0, keepdims=True)  
3 cost*=(-1/m)
```

3. Backward Pass

backward是將前面的forward透過微分的方式，從最終的output一層一層layer回推，目的是要找到修正weight的值

1. Linear backward

對應的forward是linear_forward($Z=WA+b$)的部分，根據說明部分的數學式，實作下方程式

```
1 | dW = 1/m*np.dot(dZ,A_prev.T)
2 | db = 1/m*np.sum(dZ, axis=1,
3 |                  keepdims=True)
4 | dA_prev = np.dot(W.T,dZ)
```

2. Activation backward

對應activation function的部分，因此同樣分為sigmoid與relu function兩種

1. sigmoid_backward: 根據數學式，如果要找到dZ必須要先試到sigmoid(Z)，因此先用sigmoid fn得到variable ZZ，然後將dA, ZZ分別帶入數學式中得出dZ。
2. relu_backward: relu(Z)的微分會分成 $Z>0$ ($g'(Z)=1$)及 $Z\leq 0$ ($g'(Z)=0$)兩種情形。然後因為 $dZ = dA * g'(Z)$ ，因此只有 $Z>0$ 時 $dZ=dA$ ，剩下的dZ皆是0。

3. 最後一層一層將值回推回去(linear+activation為一組)，得到各層的dW, db

4. Update parameters

透過剛剛得到各layer的dW與db值，以及自己設定的learning rate去更新layer的weight與b。

5. hyperparameter

根據建議，set layer_dim=[4,16,1], learning_rate=0.005, num_iterartion=5000。

最後得到的Accuracy = 1

Bonus Part

1. linear foward的部分與basic相同，與需要做activation function的部分(softmax function)

為了避免exponential overflow problem，經常會用「減去一個數字」的技巧。根據提示，這裡先找到各個class的最大值，將 $\exp(Z-\max)$ 即可避免overflow prblem。除此之外及如同數學式所寫，實作程式如下

```
1 | m = np.max(Z, axis=0)
2 | A = np.exp(Z-m)/np.sum(np.exp(Z-m), axis=0)
```

2. Cost Function

這裡因為有多於2 classes，不適用binary cross entropy，且因為是one-hot encode，因此使用catagorical cross entropy。根據投影片，我們可以得知下列數學式

$$-\sum_c^M \sum_i^N (Y_{\text{target}}^{i,c} \log(Y_{\text{predict}}^{i,c}))$$

並透過下方程式實作

```
1 | cost = np.sum( np.sum( (Y*np.log(AL)), axis=1,  
2 |                                     keepdims=True) , axis=0, keepdims=True)  
3 | cost*=(-1/m)
```

3. Backward Pass

根據提供網站的證明， $dZ = s - Y$ ，因此只需要找到s即可知道dZ，又知道 $S(i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$ ，因此只需要加入

```
1 | Z = cache  
2 | s = np.exp(Z)/np.sum(np.exp(Z), axis=0, keepdims=True)  
3 | dZ = (s-Y)
```

即可得到dZ。後續一樣透過dZ可以得到dW, db去更改parameter

4. hyperparameter

set layer_dims=[64,8,32,8,4], learning_rate=0.005, num_iteration=5000

最後pred_train的accuracy = 0.5981588524941126