

HW01

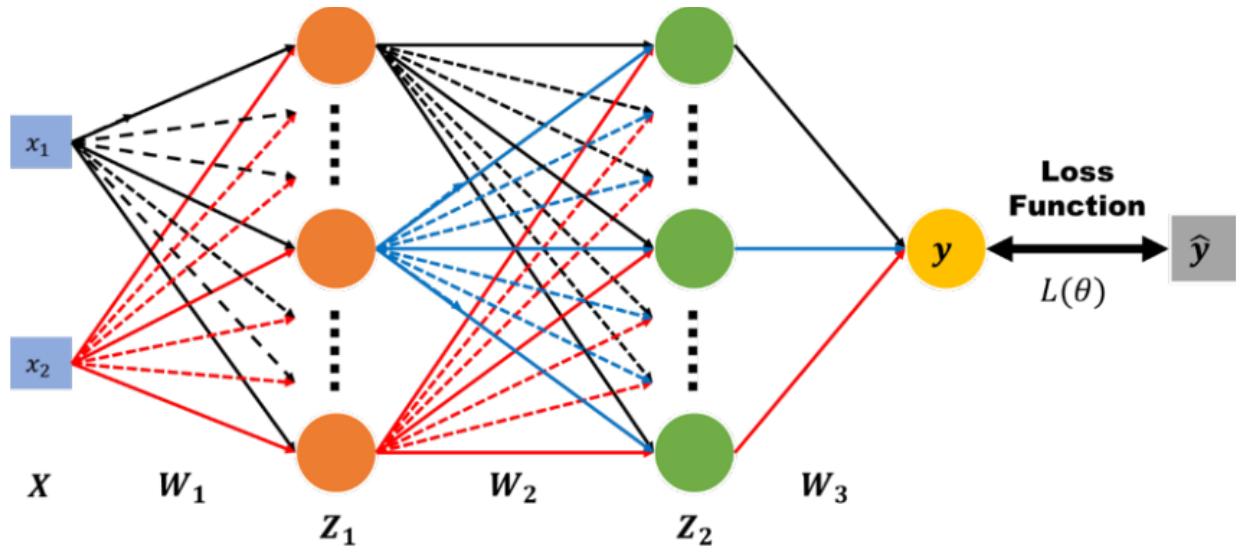
資工碩一 吳承翰 0856105

code on github: https://github.com/chiha8888/NCTU_DLP

(1) Introduction

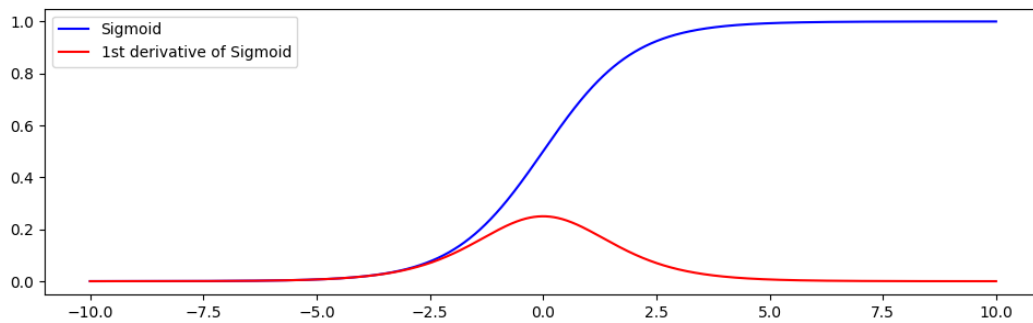
建構一個有 2 hidden layer 的 neuron network 來 classify input data.

為 logistic regression problem, loss function 為 cross entropy。



(2) Experiment Setup

(a) sigmoid function



(b) neuron network

我的 2 層 hidden layer 皆使用 10 個 neuron, learning rate 設 0.1

使用 cross entropy

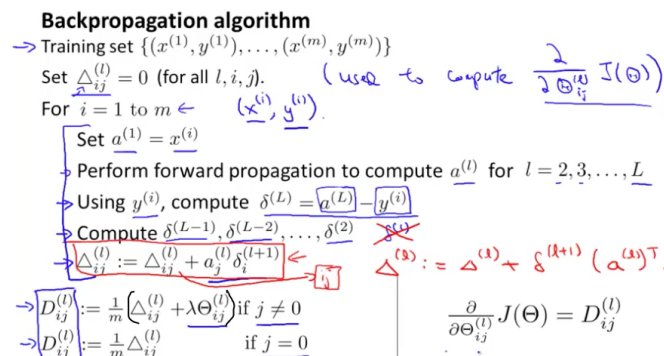
Cross entropy:

$$C(f(x^n), \hat{y}^n) = -[\hat{y}^n \ln f(x^n) + (1 - \hat{y}^n) \ln(1 - f(x^n))]$$

(c) back-propagation

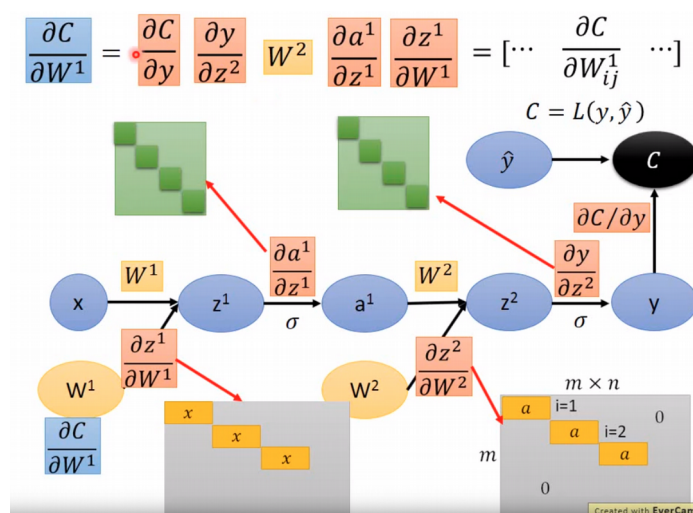
我用了 2 種實做 bp 的方式:

1.傳統 bp 方式



2.computational graph bp 方式

(reference: <https://www.youtube.com/watch?v=-yhm3WdGFok>)



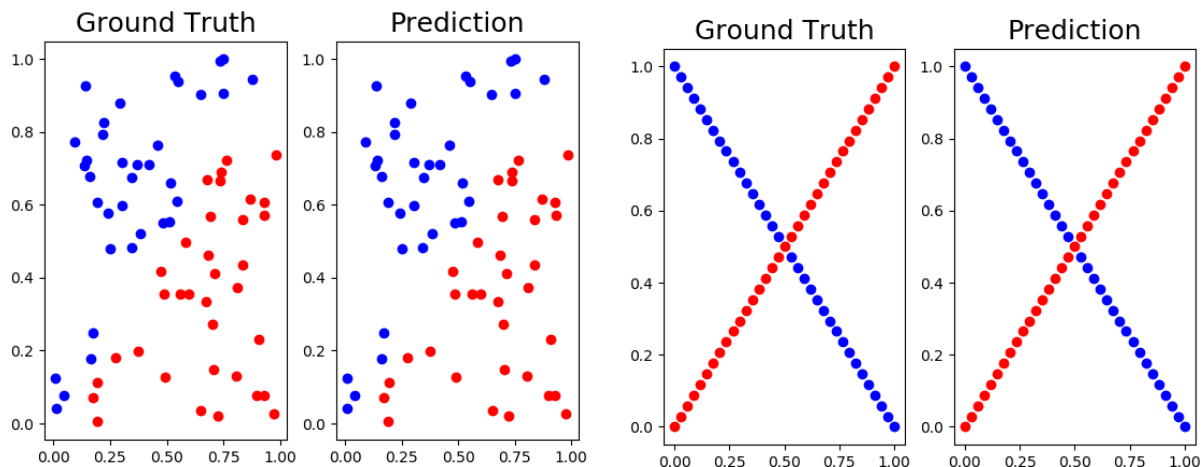
兩種方式皆可以達到收斂,

但『 1.傳統 bp 方式 』速度應該比較快也較省空間, 因為『 2.computational graph bp 方式 』有許多稀疏矩陣要相乘。

(3) Experiment result

(a) screenshot and comparison figure

兩種方法都可以達到 100% accuracy



(b) anything you want to share

我們知道：

```
a=sigmoid(z)
```

在 bp 過程中，我把 a 對 z 的微分寫成 `der_sigmoid(self.z[i])`，使得網路在 training 到十幾個 epoch 的時候就梯度爆炸了，loss 與 accuracy 皆變成 nan。
害我花了不少時間 debug。

實際上應該寫為：`der_sigmoid(self.a[i])` 或是 `der_sigmoid(sigmoid(self.z[i]))`

```
grad_z2=grad_a2*der_sigmoid(self.a[2])
```

(4) Discussion and extra experiments

我用了 2 種實做 bp 的方式：

- 1.傳統 bp 方式 (lab1_withbias.py)

- 2.computational graph bp 方式 (lab1_computationalgraph_withoutbias.ipynb)

一個有 bias 一個沒有加 bias

以 XOR dataset 來說

在到達 100% accuracy 的條件下，

我發現沒有加 bias 所需的 epoch 數僅須不到 2000，但有 bias 的 epoch 需要高達 20000 以上。

其實還蠻合理的，因為需要訓練的參數少了 `b[1],b[2],b[3]` 三個向量

但這不代表 NN 不需要 bias parameter，

只是因為這次的 dataset 很簡單，即便沒有 bias parameter 也可以訓練的非常好。

像信困難一些的 dataset 就需要 bias parameter 的加入了。