

Deep Learning and Practice - Lab0 report

Introduction

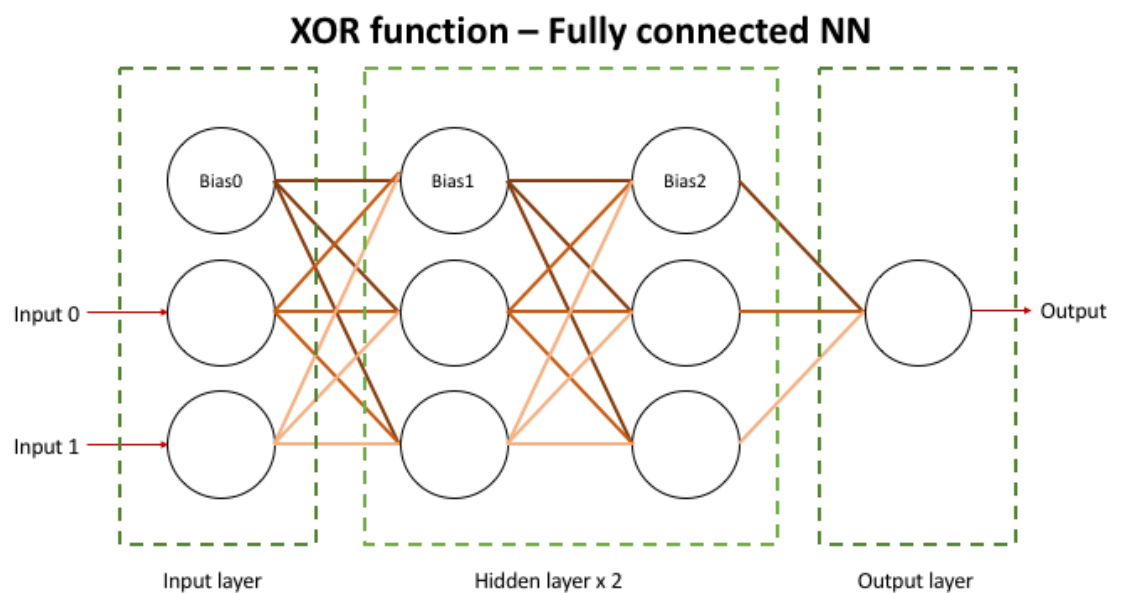
建立一個 XOR 邏輯的 Neural Network, 並且實作 backpropagation 來進行 training。

Experiment setups

A. Sigmoid function

```
def sigmoid(x):  
    return 1 / (1 + np.exp(-x))
```

B. Neural network



C. Backpropagation

目標為求出每個 weight 的變化對 Loss function 的影響，也就是 $\frac{dL}{dw}$ ，接著使用：

$$W_{i+1} = W_i + (m * \frac{dL}{dw_{i-1}} - l * \frac{dL}{dw_i})$$

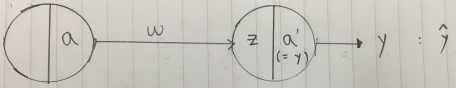
來更新各個 weight(η 為 learning rate, 並且有考慮 momentum, 將在 Discussion 討論)。

參考這篇文章 [A Step by Step Backpropagation Example](#) 的說明 (非常詳細易懂) 來實作 backpropagation, 因此每一回需要將各個 neural 的 activated output a 以及 $\frac{dL}{dz}$ (z : 丟入 activate function 前的值) 記錄下來, 這邊的話是利用 numpy 向量與矩陣來記錄, 在更新的時候進行矩陣向量的乘法即可。

下面兩張圖為兩種情況下 $\frac{dL}{dw}$ 的算法歸納：

$\frac{\partial L}{\partial w}$:

Hidden Layer \rightarrow Output Layer



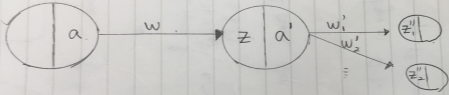
$$\begin{aligned}\frac{\partial L}{\partial w} &= a \cdot \frac{\partial L}{\partial z} \\ &= a \cdot \frac{\partial a'}{\partial z} \cdot \frac{\partial L}{\partial a'} \\ &= a \cdot a' (1-a') \cdot \frac{\partial L}{\partial a'}\end{aligned}$$

* 定義 Loss function: $\frac{1}{2} (\hat{y} - y)^2$ << from wiki >>
= backpropagation >>

$\therefore \Rightarrow a \cdot a' (1-a') \cdot (\hat{y} - y)$ *

$\frac{\partial L}{\partial w}$:

Hidden Layer \rightarrow Hidden Layer
(or Input Layer \rightarrow Hidden Layer)



$$\begin{aligned}\frac{\partial L}{\partial w} &= a \cdot \frac{\partial L}{\partial z} \\ &= a \cdot \frac{\partial a'}{\partial z} \cdot \frac{\partial L}{\partial a'} \\ &= a \cdot a' (1-a') \cdot \frac{\partial L}{\partial a'} \\ &= a \cdot a' (1-a') \cdot \sum_i w_i \frac{\partial L}{\partial z_i''}\end{aligned}$$

由於是由後往前算, 因此此項為已知。

Results

```
(lab0-env) Ricky Lab0 master± python3 lab0.py
Input data: [0. 0.] , Result: [0.00544357] Target: 0.0
Input data: [0. 1.] , Result: [0.99357837] Target: 1.0
Input data: [1. 0.] , Result: [0.99718259] Target: 1.0
Input data: [1. 1.] , Result: [0.0064005] Target: 0.0
```

Fully-connected XOR function NN.

Learning rate : 0.05

Momentum parm. : 0.5

Epoch : 30000

Hidden layer # : 2

Discussion

Momentum

在更新參數部分將上一回的 $\frac{dL}{dw_{i-1}}$ 一同納入考慮，參考現實

生活中物理慣性運動的概念，想像上一個回合的 $\frac{dL}{dw_{i-1}}$ 為原始速度，

因此每一次更新的值為 $m * \frac{dL}{dw_{i-1}} - l * \frac{dL}{dw_i}$ ， m 為一個可調參數。

此更新方法的目的為**不讓參數的更新卡在 local minimal**，有一個衡量使參數可以越過 local minimal 繼續往下前進。

Loss function def.

Loss function 的定義則是參考了 [Wiki - backpropagation](#) 中

example loss function 的定義： $L = \frac{1}{2} \|y - y'\|^2$ ，對 y'

進行偏微分即可得： $y' - y$ ，以利計算。