Search

Dashboard

MARK CHANG'S BLOG

若發生公式跑掉或無法正常顯示的情形，請在公式上按右鍵設定：math setting-> math render->SVG

- About Me
- Archive
- feeds

over 3 years ago

類神經網路 -- Backward Propagation 詳細推導過程

Introduction

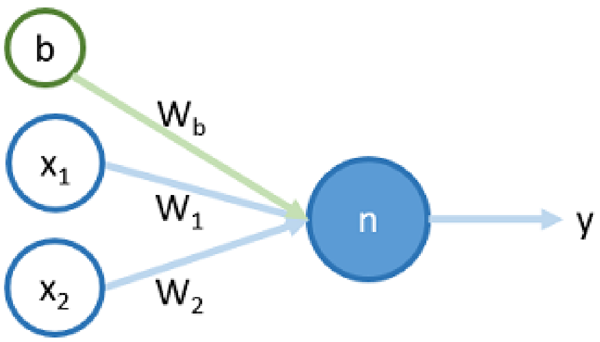
在做 Logistic Regression的時候，可以用 *gradient descent* 來做訓練，而類神經網路本身即是很多層的 *Logistic Regression* 所構成，也可以用同樣方法來做訓練。

但類神經網路在訓練過程時，需要分為兩個步驟，為： *Forward Phase* 與 *Backward Phase* 。 也就是要先從 *input* 把值傳到 *output*，再從 *output* 往回傳遞 *error* 到每一層的神經元，去更新層與層之間權重的參數。

Forward Phase

在 *Forward Phase* 時，先從 *input* 將值一層層傳遞到 *output*。

對於一個簡單的神經元 n ，如下圖 ＜圖一＞：



將一筆訓練資料 x_1, x_2 和 *bias* b 輸入到神經元 n 到輸出的過程，分成兩步，分別為 n_{in}， n_{out} ，過程如下：

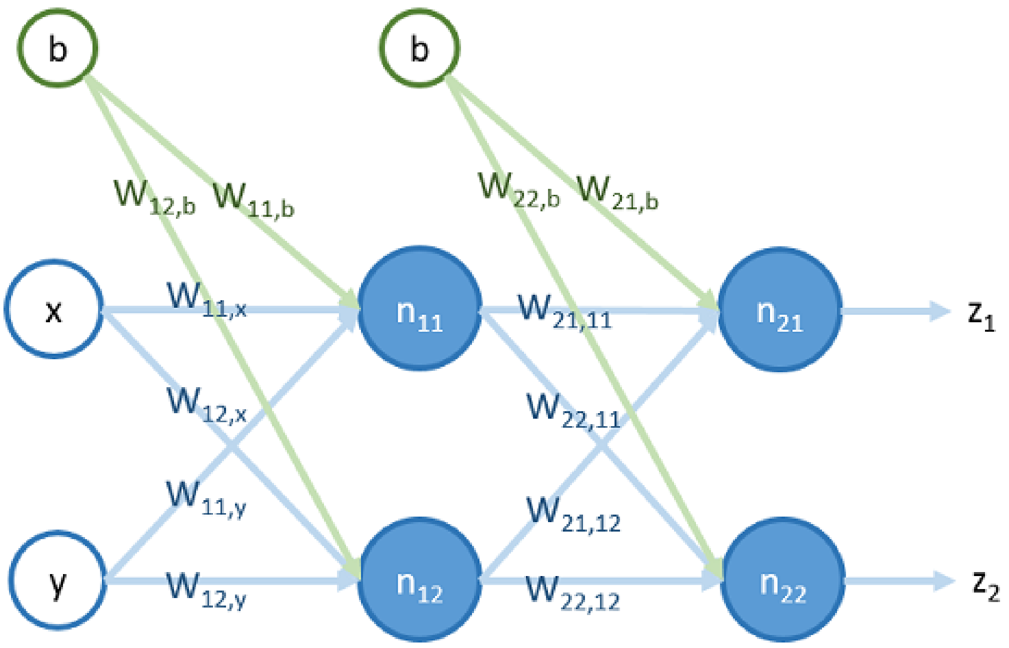
$$n_{in} = w_1 x_1 + w_2 x_2 + w_b$$
$$n_{out} = \frac{1}{1 + e^{-n_{in}}}$$

在輸入神經元時， n_{in} 先將 *input* 值和其權重作乘積。
在輸出神經元時， n_{out} 將 n_{in} 的值用 *sigmoid function* 轉成值範圍從 *0* 到 *1* 的函數。

傳遞到 n_{out} 後，可與訓練資料的答案 y 用 *cost function* 來計算其差值，並用 *backward propagation* 修正權重 w_1 、 w_2 和 w_b 。

對於一個簡單的類神經網路，共有兩層，四個神經元，如下圖＜圖二＞：

[Dashboard](#)



其值傳遞的過程如下：

1.把 x 和 y 和 $bias$ b 傳入到第一層神經元 n_{11} 及 n_{12} ：

$$\begin{aligned} n_{11(in)} &= w_{11,x}x + w_{11,y}y + w_{11,b} \\ n_{12(in)} &= w_{12,x}x + w_{12,y}y + w_{12,b} \\ n_{11(out)} &= \frac{1}{1+e^{-n_{11(in)}}} \\ n_{12(out)} &= \frac{1}{1+e^{-n_{12(in)}}} \end{aligned}$$

其中，$n_{11(in)}$ 表示傳入神經元 n_{11} 的值，而 $n_{11(out)}$ 表示傳出神經元 n_{11} 的值，而 $w_{11,x}$ 表示值從 x 傳入 n_{11} 時，所乘上的權重

2.第一層神經元將其輸出值 $n_{11(out)}$ 和 $n_{12(out)}$ 傳到第二層神經元 n_{21} 和 n_{22}：

$$\begin{aligned} n_{21(in)} &= w_{21,11}n_{11(out)} + w_{21,12}n_{12(out)} + w_{21,b} \\ n_{22(in)} &= w_{22,11}n_{11(out)} + w_{22,12}n_{12(out)} + w_{22,b} \\ n_{21(out)} &= \frac{1}{1+e^{-n_{21(in)}}} \\ n_{22(out)} &= \frac{1}{1+e^{-n_{22(in)}}} \end{aligned}$$

傳遞完後，可與訓練資料的答案 z_1 和 z_2|用 $cost\ function$ 來計算其差值，並用 $backward\ propagation$ 修正權重。

Derivation of Gradient Descent

在講解 $backward\ Phase$ 之前，先推導類神經網路的 $gradient\ descent$ 公式和 $backward\ propagation$ 的原理：

對於[<圖一>](#)中的一個簡單的神經元 n，將一筆訓練資料 x_1, x_2 傳遞到 n_{out} 所得出的值和 y 的值做比較，我們可用以下的 $cost\ function$ 來計算：

$$J = -y \times log(n_{out}) - (1-y) \times log(1-n_{out})$$

從以上 *cost function* 可得知，如果 n_{out} 和 y 都等於 *0* ，或者都等於 *1* ，則 *cost* 會是 *0* ，若 n_{out} 和 y 其中有一個是 *1* ，而另一個是 *0* ，則 *cost* 會趨近於無限大。

用 *gradient Descent* 調整 w_1 、 w_2 和 w_b 來做訓練時，可用以下公式＜公式一＞：

$$
\begin{aligned}
w_1 &\leftarrow w_1 - \eta \frac{\partial J}{\partial w_1} \\
w_2 &\leftarrow w_2 - \eta \frac{\partial J}{\partial w_2} \\
w_b &\leftarrow w_b - \eta \frac{\partial J}{\partial w_b}
\end{aligned}
$$

其中， η 為 *learning rate* ，用來控制訓練的速度。

接著要推導這個公式怎麼算，首先，將 $\frac{\partial J}{\partial w_1}$ 的微分用 *chain rule* 展開，如下 ＜公式二＞：

$$\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial n_{out}} \frac{\partial n_{out}}{\partial n_{in}} \frac{\partial n_{in}}{\partial w_1}$$

以上公式，總共有 $\frac{\partial J}{\partial n_{out}}$ 、 $\frac{\partial n_{out}}{\partial n_{in}}$ 與 $\frac{\partial n_{in}}{\partial w_1}$ 三個部份的微分要算。

1.$\frac{\partial J}{\partial n_{out}}$:

$$
\begin{aligned}
\frac{\partial J}{\partial n_{out}} &= -y \frac{\partial log(n_{out})}{\partial n_{out}} - (1-y) \frac{\partial log(1-n_{out})}{\partial n_{out}} \\
&= -\frac{y}{n_{out}} + \frac{1-y}{1-n_{out}}
\end{aligned}
$$

2.$\frac{\partial n_{out}}{\partial n_{in}}$:

$$
\begin{aligned}
\frac{\partial n_{out}}{\partial n_{in}} = \frac{\partial}{\partial n_{in}} \left(\frac{1}{1+e^{-n_{in}}}\right) = \frac{e^{-n_{in}}}{(1+e^{-n_{in}})^2} &= \frac{1}{1+e^{-n_{in}}} \frac{e^{-n_{in}}}{1+e^{-n_{in}}} \\
&= n_{out}(1-n_{out})
\end{aligned}
$$

3.$\frac{\partial n_{in}}{\partial w_1}$:

$$
\begin{aligned}
\frac{\partial n_{in}}{\partial w_1} &= \frac{\partial}{\partial w_1} (w_1 x_1 + w_2 x_2 + w_b) \\
&= x_1
\end{aligned}
$$

代入以上三個結果到[＜公式二＞]，可得出 $\frac{\partial J}{\partial w_1}$ 的值，如下：

$$
\begin{aligned}
\frac{\partial J}{\partial w_1} &= \frac{\partial J}{\partial n_{out}} \frac{\partial n_{out}}{\partial n_{in}} \frac{\partial n_{in}}{\partial w_1} \\
&= \left(-\frac{y}{n_{out}} + \frac{1-y}{1-n_{out}}\right) n_{out}(1-n_{out}) x_1 \\
&= (n_{out} - y) x_1
\end{aligned}
$$

同理可得出 $\frac{\partial J}{\partial w_2}$ 與 $\frac{\partial J}{\partial w_b}$ 的值，分別為： Dashboard

$$\frac{\partial J}{\partial w_2} = \frac{\partial J}{\partial n_{out}} \frac{\partial n_{out}}{\partial n_{in}} \frac{\partial n_{in}}{\partial w_2} = (n_{out} - y)x_2$$
$$\frac{\partial J}{\partial w_b} = \frac{\partial J}{\partial n_{out}} \frac{\partial n_{out}}{\partial n_{in}} \frac{\partial n_{in}}{\partial w_b} = (n_{out} - y)$$

其中， $\frac{\partial n_{in}}{\partial w_b}$ 的結果為：

$$\frac{\partial n_{in}}{\partial w_b} = \frac{\partial}{\partial w_b}(w_1x_1 + w_2x_2 + w_b) = 1$$

將 $\frac{\partial J}{\partial w_1}$ 、 $\frac{\partial J}{\partial w_2}$ 和 $\frac{\partial J}{\partial w_b}$ 的結果代入＜公式一＞，得出：

$$w_1 \leftarrow w_1 - \eta(n_{out} - y)x_1$$
$$w_2 \leftarrow w_2 - \eta(n_{out} - y)x_2$$
$$w_b \leftarrow w_b - \eta(n_{out} - y)$$

Derivation of Backward Propagation

若要推導超過一層的類神經網路的 *gradient descent* 公式，就要用到 *backward propagation* 。

對於＜圖二＞中的一個簡單的類神經網路，它的 *cost function* 如下：

$$J = -(z_1 \times log(n_{21(out)}) + (1 - z_1) \times log(1 - n_{21(out)}) + z_2 \times log(n_{22(out)}) + (1 - z_2) \times log(1 - n_{22(out)}))$$

對於最後一層與倒數第二層之間的權重改變，可用 *gradient descent* ，如下＜公式三＞：

$$w_{21,11} \leftarrow w_{21,11} - \eta \frac{\partial J}{\partial w_{21,11}}$$
$$w_{21,12} \leftarrow w_{21,12} - \eta \frac{\partial J}{\partial w_{21,12}}$$
$$w_{21,b} \leftarrow w_{21,b} - \eta \frac{\partial J}{\partial w_{21,b}}$$

可用先前推導出單一神經元時的微分結果，得出：

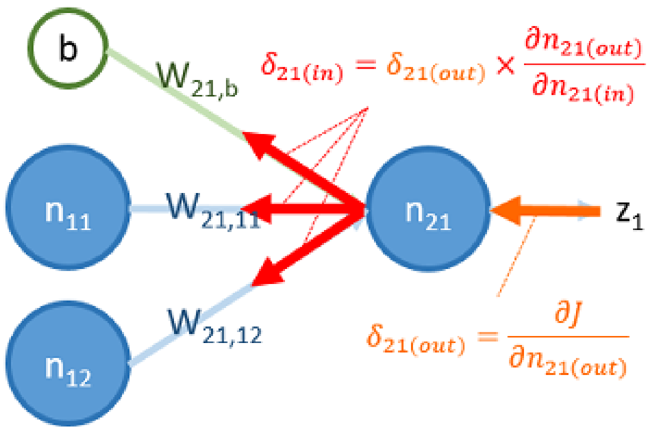
$$\frac{\partial J}{\partial w_{21,11}} = \frac{\partial J}{\partial n_{21(out)}} \frac{\partial n_{21(out)}}{\partial n_{21(in)}} \frac{\partial n_{21(in)}}{\partial w_{21,11}} = (n_{21(out)} - z_1)n_{11(out)}$$
$$\frac{\partial J}{\partial w_{21,12}} = \frac{\partial J}{\partial n_{21(out)}} \frac{\partial n_{21(out)}}{\partial n_{21(in)}} \frac{\partial n_{21(in)}}{\partial w_{21,12}} = (n_{21(out)} - z_1)n_{12(out)}$$
$$\frac{\partial J}{\partial w_{21,b}} = \frac{\partial J}{\partial n_{21(out)}} \frac{\partial n_{21(out)}}{\partial n_{21(in)}} \frac{\partial n_{21(in)}}{\partial w_{21,b}} = (n_{21(out)} - z_1)$$

同理可求出 $w_{22,11}$ 、 $w_{22,12}$ 和 $w_{22,b}$ 相對應的公式。

在要推導更往前一層的權重變化公式之前，先觀察以上公式，發現它們有共同的部分： $n_{21(out)} - z_1$ ，可以用 $\delta_{21(in)}$ 來表示這個值，即：

$$\delta_{21(in)} = \frac{\partial J}{\partial n_{21(out)}} \frac{\partial n_{21(out)}}{\partial n_{21(in)}} = n_{21(out)} - z_1$$
$$\frac{\partial J}{\partial w_{21,11}} = \delta_{21(in)} n_{11(out)}$$
$$\frac{\partial J}{\partial w_{21,12}} = \delta_{21(in)} n_{12(out)}$$
$$\frac{\partial J}{\partial w_{21,b}} = \delta_{21(in)}$$

$\delta_{21(in)}$ 的物理意義如下圖所示：



圖中， $\delta_{21(out)}$ 是 J 在神經元 n_{21} 輸出點的微分值，可以把 $\delta_{21(in)}$ 看成是 $\delta_{21(out)}$ 從神經元 $n_{21|}$ 的輸出點往回傳到輸入點，即乘上 $\frac{\partial n_{21(out)}}{\partial n_{21(in)}}$ 。因此，這過程又稱為 *backward propagation* 。

將 $\delta_{21(in)}$ 置換到<公式三>，得出這一層推導的最後結果：

$$w_{21,11} \leftarrow w_{21,11} - \eta \delta_{21(in)} n_{11(out)}$$
$$w_{21,12} \leftarrow w_{21,12} - \eta \delta_{21(in)} n_{12(out)}$$
$$w_{21,b} \leftarrow w_{21,b} - \eta \delta_{21(in)}$$

同理， $w_{22,11}, w_{22,12}, w_{22,b}$ 的 *gradient descent* 公式，也可用相同方法推導出來：

$$w_{22,11} \leftarrow w_{22,11} - \eta \delta_{22(in)} n_{11(out)}$$
$$w_{22,12} \leftarrow w_{22,12} - \eta \delta_{22(in)} n_{12(out)}$$
$$w_{22,b} \leftarrow w_{22,b} - \eta \delta_{22(in)}$$

再來，要推導更往前一層的權重變化公式，要用 *gradient descent* <公式四>：

$$w_{11,x} \leftarrow w_{11,x} - \eta \frac{\partial J}{\partial w_{11,x}}$$
$$w_{11,y} \leftarrow w_{11,y} - \eta \frac{\partial J}{\partial w_{11,y}}$$
$$w_{11,b} \leftarrow w_{11,b} - \eta \frac{\partial J}{\partial w_{11,b}}$$

舉 $w_{11,x}$ 為例，用 *chain rule* 求出 $\frac{\partial J}{\partial w_{11,x}}$ 的值，如下<公式五>：

Dashboard

$$
\begin{aligned}
\frac{\partial J}{\partial w_{11,x}} &= \frac{\partial J}{\partial n_{21(out)}}\frac{\partial n_{21(out)}}{\partial w_{11,x}} + \frac{\partial J}{\partial n_{22(out)}}\frac{\partial n_{22(out)}}{\partial w_{11,x}} \\
&= \frac{\partial J}{\partial n_{21(out)}}\frac{\partial n_{21(out)}}{\partial n_{21(in)}}\frac{\partial n_{21(in)}}{\partial n_{11(out)}}\frac{\partial n_{11(out)}}{\partial n_{11(in)}}\frac{\partial n_{11(in)}}{\partial w_{11,x}} + \frac{\partial J_2}{\partial n_{22(out)}}\frac{\partial n_{22(out)}}{\partial n_{22(in)}}\frac{\partial n_{22(in)}}{\partial n_{11(out)}}\frac{\partial n_{11(out)}}{\partial n_{11(in)}}\frac{\partial n_{11(in)}}{\partial w_{11,x}} \\
&= \Big(\frac{\partial J}{\partial n_{21(out)}}\frac{\partial n_{21(out)}}{\partial n_{21(in)}}\frac{\partial n_{21(in)}}{\partial n_{11(out)}} + \frac{\partial J}{\partial n_{22(out)}}\frac{\partial n_{22(out)}}{\partial n_{22(in)}}\frac{\partial n_{22(in)}}{\partial n_{11(out)}}\Big)\frac{\partial n_{11(out)}}{\partial n_{11(in)}}\frac{\partial n_{11(in)}}{\partial w_{11,x}}
\end{aligned}
$$

其中，$\frac{\partial n_{21(in)}}{\partial n_{11(out)}}$ 、$\frac{\partial n_{22(in)}}{\partial n_{11(out)}}$ 、$\frac{\partial n_{11(out)}}{\partial n_{11(in)}}$ 和 $\frac{\partial n_{11(in)}}{\partial w_{11,x}}$ 這四項的值分別為：

$$
\begin{aligned}
\frac{\partial n_{21(in)}}{\partial n_{11(out)}} &= \frac{\partial}{\partial n_{11(out)}}(w_{21,11}n_{11(out)} + w_{21,12}n_{12(out)} + w_{21,b}) = w_{21,11} \\
\frac{\partial n_{22(in)}}{\partial n_{11(out)}} &= \frac{\partial}{\partial n_{11(out)}}(w_{22,11}n_{11(out)} + w_{22,12}n_{12(out)} + w_{22,b}) = w_{22,11} \\
\frac{\partial n_{11(out)}}{\partial n_{11(in)}} &= \frac{\partial}{\partial n_{11(in)}}\Big(\frac{1}{1+e^{-n_{11(in)}}}\Big) = n_{11(out)}(1 - n_{11(out)}) \\
\frac{\partial n_{11(in)}}{\partial w_{11,x}} &= \frac{\partial}{\partial w_{11,x}}(w_{11,x}x + w_{11,y}y + w_{11,b}) = x
\end{aligned}
$$

再代入這些值與之前推導出的 $\frac{\partial J}{\partial n_{21(out)}}\frac{\partial n_{21(out)}}{\partial n_{21(in)}}$ 和 $\frac{\partial J}{\partial n_{22(out)}}\frac{\partial n_{22(out)}}{\partial n_{22(in)}}$ 的值到＜公式五＞，可求出 $\frac{\partial J}{\partial w_{11,x}}$ 為：

$$
\frac{\partial J}{\partial w_{11,x}} = ((n_{21(out)} - z_1)w_{21,11} + (n_{22(out)} - z_2)w_{22,11})n_{11(out)}(1 - n_{11(out)})x
$$

同理，可求出 $\frac{\partial J}{\partial w_{11,y}}$ 和 $\frac{\partial J}{\partial w_{11,b}}$ 的值分別為：

$$
\begin{aligned}
\frac{\partial J}{\partial w_{11,y}} &= ((n_{21(out)} - z_1)w_{21,11} + (n_{22(out)} - z_2)w_{22,11})n_{11(out)}(1 - n_{11(out)})y \\
\frac{\partial J}{\partial w_{11,b}} &= ((n_{21(out)} - z_1)w_{21,11} + (n_{22(out)} - z_2)w_{22,11})n_{11(out)}(1 - n_{11(out)})
\end{aligned}
$$

如同前一層所推導的，以上公式也有相同部分，也可以用 $\delta_{11(in)}$ 來簡化它們，如下：

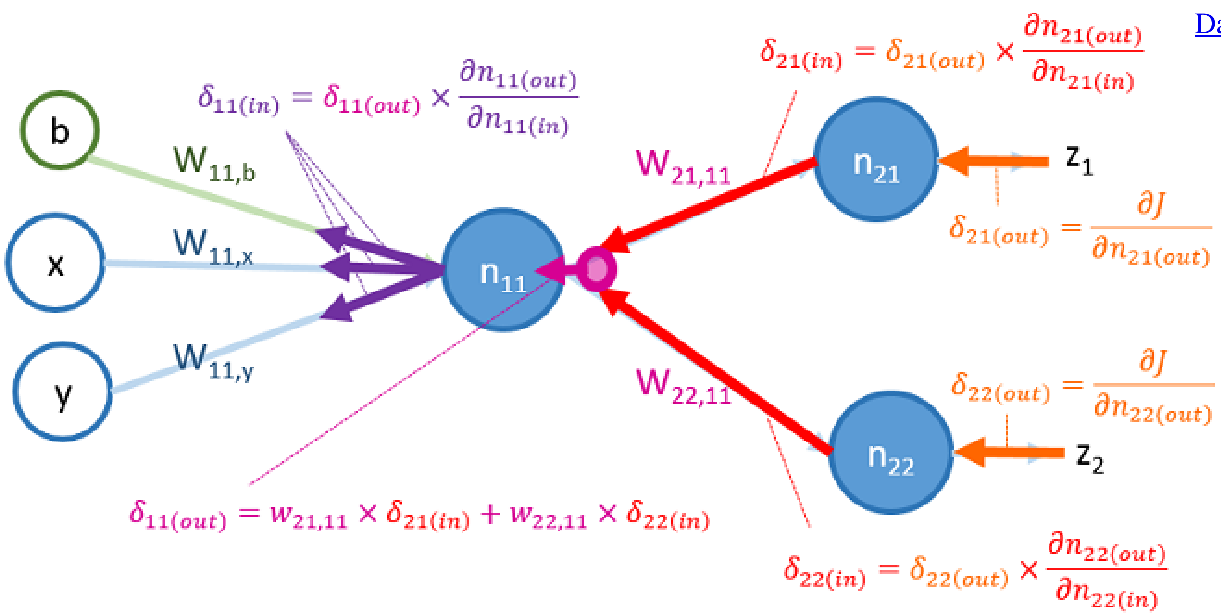
$$
\begin{aligned}
\delta_{11(in)} &= ((n_{21(out)} - z_1)w_{21,11} + (n_{22(out)} - z_2)w_{22,11})n_{11(out)}(1 - n_{11(out)}) \\
\frac{\partial J}{\partial w_{11,x}} &= \delta_{11(in)}x \\
\frac{\partial J}{\partial w_{11,y}} &= \delta_{11(in)}y \\
\frac{\partial J}{\partial w_{11,b}} &= \delta_{11(in)}
\end{aligned}
$$

可把 $\delta_{11(in)}$ 用後面層傳回來的的 δ 來表示，如下：

$$
\begin{aligned}
\delta_{11(out)} &= w_{21,11}\delta_{21(in)} + w_{22,11}\delta_{22(in)} \\
\delta_{11(in)} &= \delta_{11(out)}n_{11(out)}(1 - n_{11(out)}) = \delta_{11(out)}\frac{\partial n_{11(out)}}{\partial n_{11(in)}}
\end{aligned}
$$

這些 δ 的物理意義如下圖所示：

Dashboard



從圖中可以看到， $\delta_{11(out)}$ 是 由 $\delta_{21(in)}$ 和 $\delta_{22(in)}$ 往反方向傳遞，再乘上其權重 $w_{21,11}$ 與 $w_{22,11}$ 所得出的 。

將 $\delta_{11(in)}$ 置換到＜公式四＞，得出這一層推導的最後結果：

$$
\begin{aligned}
w_{11,x} &\leftarrow w_{11,x} - \eta\delta_{11(in)}x \\
w_{11,y} &\leftarrow w_{11,y} - \eta\delta_{11(in)}y \\
w_{11,b} &\leftarrow w_{11,b} - \eta\delta_{11(in)}
\end{aligned}
$$

同理， $w_{12,x}, w_{12,y}, w_{12,b}$ 的 *gradient descent* 的公式，也可用相同方法推導出來：

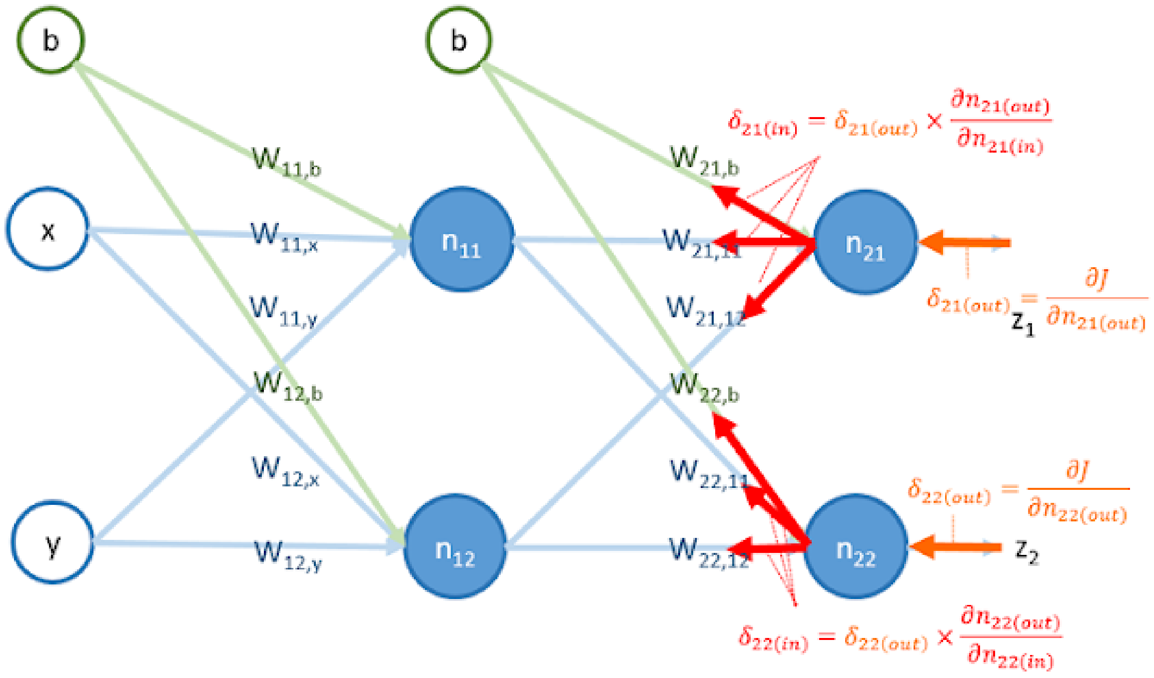
$$
\begin{aligned}
w_{12,x} &\leftarrow w_{12,x} - \eta\delta_{12(in)}x \\
w_{12,y} &\leftarrow w_{12,y} - \eta\delta_{12(in)}y \\
w_{12,b} &\leftarrow w_{12,b} - \eta\delta_{12(in)}
\end{aligned}
$$

Backward Phase

backward phase 要做的即是 *backward propagation* ，也就是從 *output* 把 δ 算出來，並更新權重 w ，再把 δ往回傳一層，再更新那層的權重 w，這樣一直傳下去直到 *input* 。

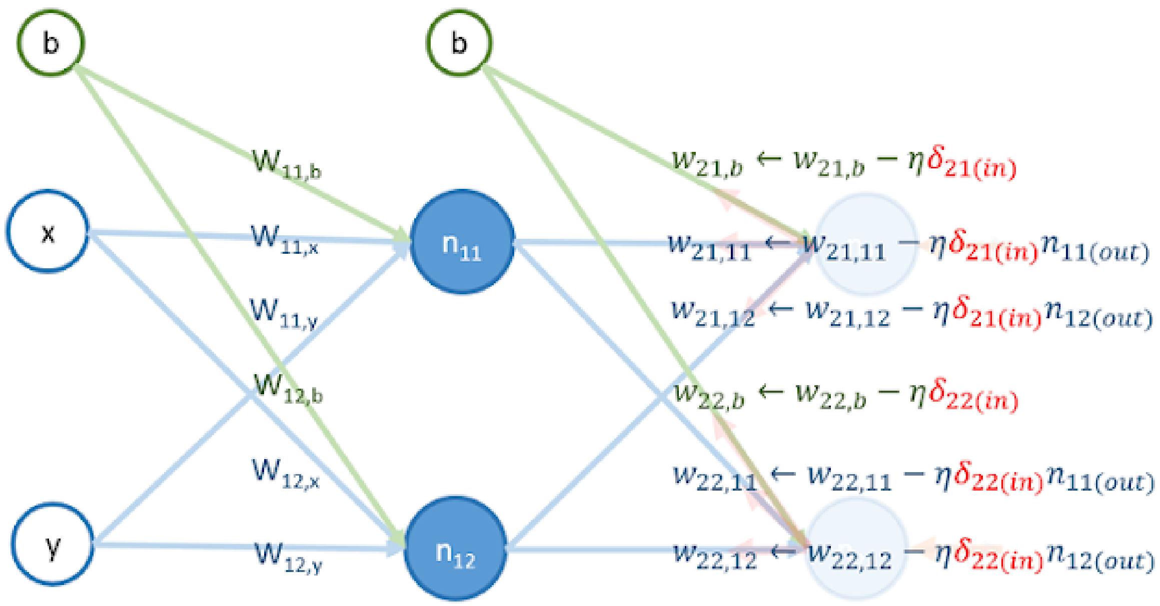
首先，把 $\delta_{21(in)}$ 和 $\delta_{22(in)}$ 算出來：

$$
\begin{aligned}
\delta_{21(in)} &= n_{21(out)} - z_1 \\
\delta_{22(in)} &= n_{22(out)} - z_2
\end{aligned}
$$



再來，用 $\delta_{21(in)}$ 和 $\delta_{22(in)}$ 更新以下權重的值：

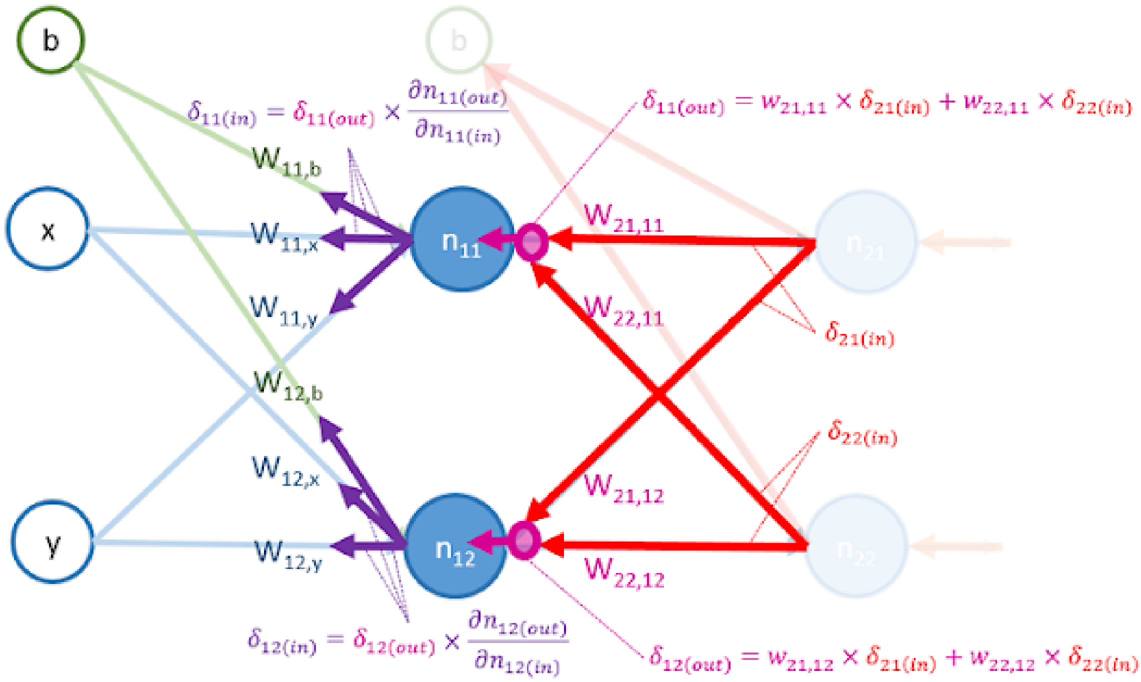
$$
\begin{aligned}
w_{21,11} &\leftarrow w_{21,11} - \eta\delta_{21(in)}n_{11(out)} \\
w_{21,12} &\leftarrow w_{21,12} - \eta\delta_{21(in)}n_{12(out)} \\
w_{21,b} &\leftarrow w_{21,b} - \eta\delta_{21(in)} \\
w_{22,11} &\leftarrow w_{22,11} - \eta\delta_{22(in)}n_{11(out)} \\
w_{22,12} &\leftarrow w_{22,12} - \eta\delta_{22(in)}n_{12(out)} \\
w_{22,b} &\leftarrow w_{22,b} - \eta\delta_{22(in)}
\end{aligned}
$$



再來，把 $\delta_{21(in)}$ 和 $\delta_{22(in)}$ 乘上權重，算出 $\delta_{11(in)}$ 和 $\delta_{12(in)}$的值：

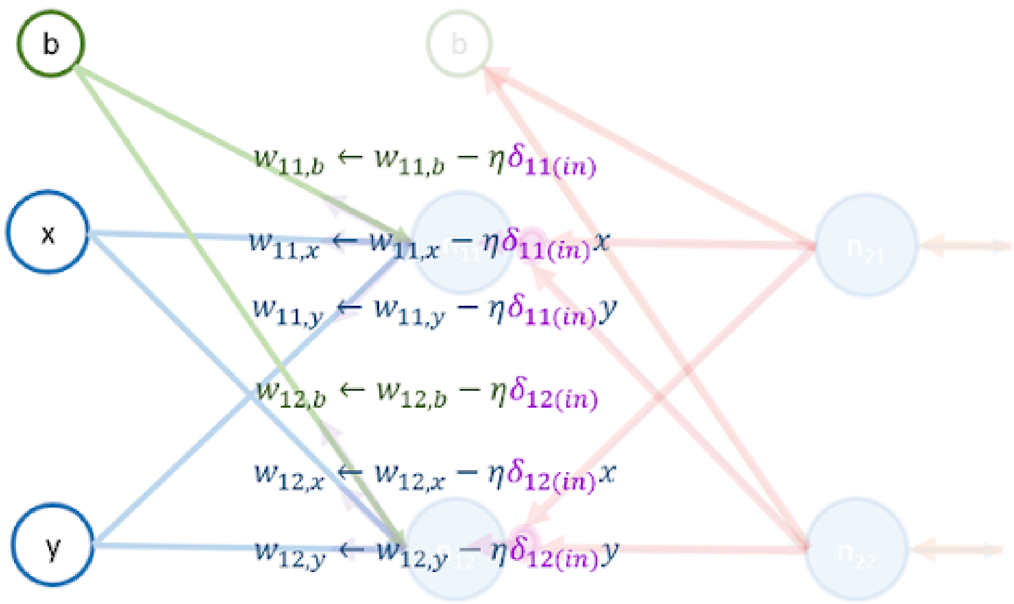
$$
\begin{aligned}
\delta_{11(in)} &= (w_{21,11}\delta_{21(in)} + w_{22,11}\delta_{22(in)})n_{11(out)}(1 - n_{11(out)}) \\
\delta_{12(in)} &= (w_{21,12}\delta_{21(in)} + w_{22,12}\delta_{22(in)})n_{12(out)}(1 - n_{12(out)})
\end{aligned}
$$

Dashboard



最後，用 $\delta_{11(in)}$ 和 $\delta_{12(in)}$ 更新以下權重的值：

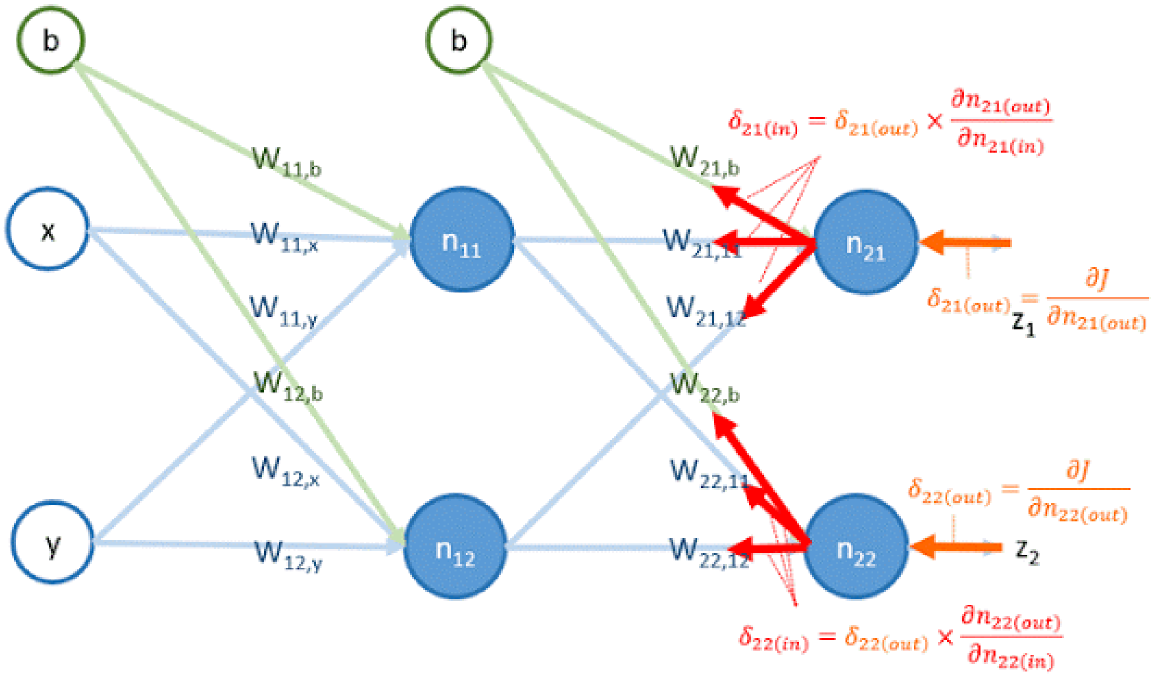
$$\begin{aligned}
w_{11,x} &\leftarrow w_{11,x} - \eta\delta_{11(in)}x \\
w_{11,y} &\leftarrow w_{11,y} - \eta\delta_{11(in)}y \\
w_{11,b} &\leftarrow w_{11,b} - \eta\delta_{11(in)} \\
w_{12,x} &\leftarrow w_{12,x} - \eta\delta_{12(in)}x \\
w_{12,y} &\leftarrow w_{12,y} - \eta\delta_{12(in)}y \\
w_{12,b} &\leftarrow w_{12,b} - \eta\delta_{12(in)}
\end{aligned}$$



更新完後，即結束了在資料 x, y 上的這一輪訓練。

以下為整個過程的動畫版：

Dashboard



Reference

本文參考 coursera 課程 Andrew Ng. Machine Learning
https://www.coursera.org/course/ml

← 類神經網路 -- Hierarchical Probabilistic Neural Network Language Model (Hierarchical Softmax) 類神經網路 -- Recurrent Neural Network →

- neural_network
- back_propagation

- May 28, 2015 15:47
- Permalink
- 6 Comments