

Change

SUCCESS BELONGS TO THE ONE WITH CLEAR AND LONG-TERM GOALS!

[HOME](#)[CONTACT](#)[GALLERY](#)[SUBSCRIBE](#)

BP神經網路演算法推導

2012-12-12 21:50 錢吉 閱讀(23129) 評論(1) 編輯 收藏 舉報

前言：自己動手推導了一下經典的前向回饋神經網路的演算法公式，記錄一下。由於暫時沒有資料可以用作測試，程式沒有實現並驗證。以後找到比較好的數據，再實現。

一：演算法推導

神經網路透過模擬人的神經元活動，來建構分類器。它的基本組成單元稱為“神經元”，離線情況下如果輸入大於某值時，設定神經元處於興奮狀態，產生輸出，否則不響應。而這個輸入來自於所有其它的神經元。而神經元的反應函數有多種(需要滿足可微，這種簡單的函數可以擬合任何非線性函數)，本文選擇sigmoid函數。關於基礎知識在此不在多說，這裡主要介紹一下BP神經網路，並推導出權值和閾值的更新公式。

BP網路透過前向回饋誤差來改變權值和閾值，當網路的輸出與實際期望的輸出均方誤差低於某個閾值或學習次數滿足某個條件時，訓練完成。首先給出輸出均方誤差的公式：

About

暱稱：[錢吉](#)園齡：[12年4個月](#)粉絲：[70](#)追蹤：[37](#)[+加關注](#)

SEARCH

最新評論

[Re:尾遞歸和編譯器最佳化](#)

Hi，我寫了一篇關於遞歸的總結，歡迎指教，

-- yiifburj

[Re:求所有最大公共子序列的演算法實現](#)

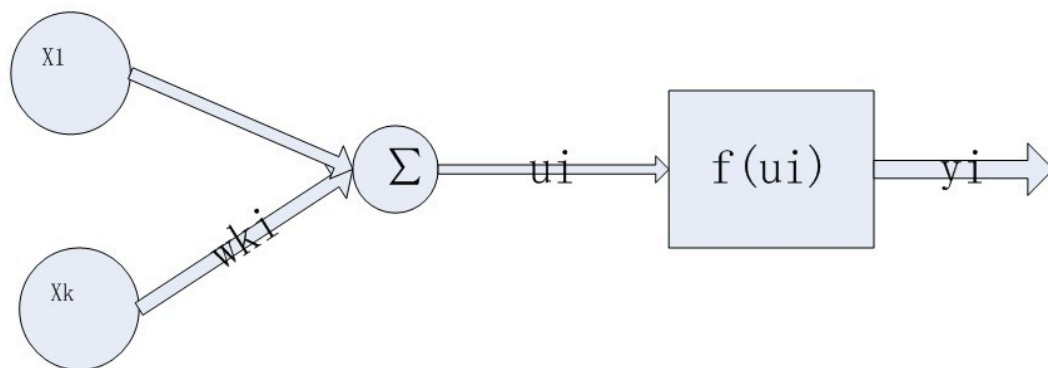
大佬的網盤連結8年了還在生效!佩服!

-- 菜鳥qiz

$$E_A = \frac{1}{2} \sum_{p=1}^P \sum_{i=0}^{m-1} (d_i^{(p)} - y_i^{(p)})^2$$

其中， m 是輸出節點數， p 是訓練樣本數。我們就是以最小化這個 E_A 為目標更新網路參數。這裡以3層網路為例進行推導，即一個輸入層、隱含層、輸出層。最小化目標函數採用梯度下降的方法。

1. 隱含層到輸出層的權值更新



K 為隱含層節點數， i 為輸出節點數。權值更新公式：

$$w_{ki} = w_{ki}' - \eta \nabla E(w_{ki})$$

[Re:如何利用迴圈代替遞歸以防止棧溢位\(譯\)](#)

5 替代過程的幾個簡單例子-線性遞歸-74行應該是returnVal = returnVal *1; -- 楊某某

[Re:GetWindowRect和GetClientRect的差異詳解](#)

謝謝分享，學習了。 -- 雪域迷影

[Re:GetWindowRect和GetClientRect的差異詳解](#)

謝謝~ -- 秦漢思源

日曆

< 2012年12月 >						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	<u>7</u>	<u>8</u>
9	10	11	<u>12</u>	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

我的標籤

[c/c++ \(19\)](#)

[matlab \(8\)](#)

[資料結構與演算法\(7\)](#)

[navigation \(4\)](#)

[翻譯\(3\)](#)

[基礎演算法\(3\)](#)

[Windows \(3\)](#)

[bat處理\(3\)](#)

[雜碎\(2\)](#)

[機器學習\(2\)](#)

隨筆檔案

[2016年12月\(1\)](#)

[2014年7月\(6\)](#)

[2014年6月\(2\)](#)

[2014年3月\(1\)](#)

[2013年12月\(1\)](#)

[2013年11月\(3\)](#)

[2013年8月\(3\)](#)

[2013年7月\(9\)](#)

[2013年6月\(5\)](#)

[2013年5月\(2\)](#)

[2013年4月\(2\)](#)

[2013年3月\(1\)](#)

[2012年12月\(7\)](#)

[2012年11月\(3\)](#)

[2012年10月\(9\)](#)

[更多](#)

IT雜談

[外刊IT評論網](#)

[虎嗅網](#)

其中 w_{ki} 表示第 k 個隱含節點到第 i 個輸出節點之間的權重， η 為學習率，是一個使得解快

速收斂的學習因子， $\nabla E(w_{ki})$ 為EA關於 w_{ki} 的梯度。即：

$$\nabla E(w_{ki}) = \frac{\partial E_A}{\partial w_{ki}} = \sum_{p=1}^P \frac{\partial E_A}{\partial y_i^{(p)}} \frac{\partial y_i^{(p)}}{\partial u_i^{(p)}} \frac{\partial u_i^{(p)}}{\partial w_{ki}}$$

因為：

$$u_i^{(p)} = \sum_{k=1}^K w_{ki} * x_k^{(p)}; y_i^{(p)} = \frac{1}{1 + e^{-u_i^{(p)}}}$$

，其中， σ 為激活函數

$$y = \frac{1}{1 + e^{-x}}$$

對 x 求導後， $y' = y(1-y)$ ，所以再進一步求：

$$\frac{\partial E_A}{\partial w_{ki}} = - \sum_{p=1}^P (d_i^{(p)} - y_i^{(p)}) * y_i^{(p)} * (1 - y_i^{(p)}) * x_k^{(p)}$$

這就是 w_{ki} 的梯度值。記：

$$\Delta w_{ki} = -\eta \nabla E(w_{ki})$$

，表示為權值的增量，則權值更新可寫成：

[更多](#)

隨筆分類

[c/c++\(21\)](#)

[lua\(1\)](#)

[matlab\(6\)](#)

[perl\(2\)](#)

[windows\(7\)](#)

[車載導航\(4\)](#)

[累積\(3\)](#)

[設計模式\(1\)](#)

[資料結構與演算法\(5\)](#)

[語音辨識演算法\(4\)](#)

[職場感悟\(1\)](#)

我的足跡

[我的github](#)

[開源中國博客](#)

[chinaunix](#)

[csdn](#)

學習站點

[codeproject](#)

[stackoverflow](#)

[GitHub](#)

[Apache Hadoop](#)

[開源中國](#)

[Taiwan Hadoop Forum](#)

高手博客

[ifreecoding](#)

[Hawstein's blog\(Cracking the coding interview-c++\)](#)

[百度搜尋研發部博客](#)

[IBM developer](#)

[csdn-July演算法之道](#)

[劉未鵬](#)

[何海濤-程式設計師面試](#)

[董的部落格-大數據開發](#)

[Free Mind的博客](#)

[阿里核心團隊博客](#)

資源

[it-ebooks](#)

積分與排名

積分- 94235

排名- 16060

$$w_{ki} = w_{ki}' + \Delta w_{ki}$$

其中增量:

$$\Delta w_{ki} = \eta \sum_{p=1}^P (d_i^{(p)} - y_i^{(p)}) * y_i^{(p)} * (1 - y_i^{(p)}) * x_k^{(p)}$$

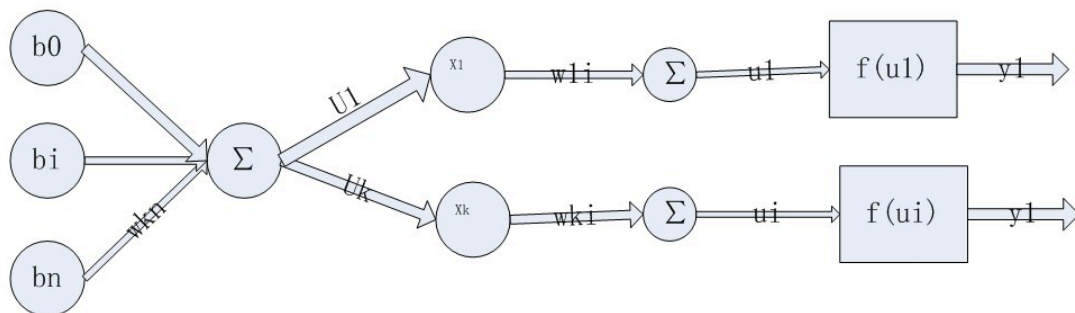
记 $\xi_{ki} = (d_i^{(p)} - y_i^{(p)}) * y_i^{(p)} * (1 - y_i^{(p)})$ ，则

$$\Delta w_{ki} = \eta \sum_{p=1}^P \xi_{ki} * x_k^{(p)}, \quad w_{ki} = w_{ki}' + \eta \sum_{p=1}^P \xi_{ki} * x_k^{(p)}。$$

所以根據這個式子我們就可以更新權值了。

2.輸入層到隱含層的權值更新

同理，誤差關於權值的梯度可透過以下式子求解，與上面有點不同的是：輸入層與隱含節點之間的權值將影響所有的隱含節點到輸出層之間的值，所以此時的權值梯度資訊應該會對誤差在隱含層與輸出層之間產生累加效應。廢話不多說，直接上圖公式更明了：



$$\begin{aligned}
 \nabla E(w_{kn}) &= \frac{\partial E_A}{\partial w_{kn}} = \sum_{p=1}^P \sum_{i=1}^{m-1} \frac{\partial E^{(p)}}{\partial y_i^{(p)}} \frac{\partial y_i^{(p)}}{\partial u_i^{(p)}} \frac{\partial u_i^{(p)}}{\partial x_k^{(p)}} \frac{\partial x_k^{(p)}}{\partial u_k^{(p)}} \frac{\partial u_k^{(p)}}{\partial w_{kn}} \\
 &= -\sum_{p=1}^P \sum_{i=1}^{m-1} (d_i^{(p)} - y_i^{(p)}) * f'(u_i^{(p)}) * w_{ki} * u_k^{(p)} * (1 - u_k^{(p)}) * b_n^{(p)} \\
 &= -\sum_{p=1}^P \sum_{i=1}^{m-1} \xi_{ki}^{(p)} * w_{ki} * u_k^{(p)} * (1 - u_k^{(p)}) * b_n^{(p)} \\
 &= -\sum_{p=1}^P \xi_{kn}^{(p)} * b_n^{(p)}
 \end{aligned}$$

其中，

$$\xi_{kn} = \sum_{i=1}^{m-1} \xi_{ki}^{(p)} * w_{ki} * u_k^{(p)} * (1 - u_k^{(p)})$$

所以這一層的權值增量也可以寫成這樣的形式：

$$w_{kn} = w_{kn}' + \eta \sum_{p=1}^P \xi_{kn}^{(p)} * b_n^{(p)}$$

參考：

<http://www.cnblogs.com/hellope/archive/2012/07/05/2577814.html>

<http://baike.baidu.com/view/1753676.htm>

- 分類 [語音辨識演算法](#)
- 標籤 [機器學習](#)

好文要頂

關注我

收藏該文

微信分享



錢吉

粉絲- 70 追蹤- 37

0

0

+加關注

[升級成為會員](#)

« 上一篇：[求所有最大公共子序列的演算法實作](#)

» 下一篇：[圖論：最短路徑搜尋--Dijkstra演算法\(c程式碼實作\)](#)

會員力量，點亮園子希望

[刷新頁面](#) [返回頂部](#)

登入後才能查看或發表評論，立即[登入](#)或 [逛逛](#)部落格園首頁

【推薦】超值焕新月，阿里雲2核2G雲端伺服器99元/年，立即搶購

【推薦】園子週邊第二季：更大的滑鼠墊，沒有logo的滑鼠墊

【推薦】阿里雲雲市場聯合博客園推出開發者商店，歡迎關注

【推薦】會員力量，點亮園子希望，期待您升級成為園子會員



編輯推薦：

- 程式設計師天天CURD，怎麼才能成長，職涯發展的思考？
- 小程序中使用lottie 動畫| 踩坑經驗分享
- Windows下綁定執行緒到指定的CPU核心
- C++裡也有菱形運算子？
- [Nano Framework ESP32篇] 使用LCD 螢幕

閱讀排行：

- C#的基於.net framework的Dll模組程式設計（五） - 程式手把手系列文章
- 程式設計師天天CURD，怎麼才能成長，職業發展的思考？
- .Net 8.0 下的新RPC，IceRPC之試試的新玩法"打洞"
- .NET CORE 完美支援AOT 的ORM SqlSugar 教程
- 開源文檔預覽項目kkFileView (9.9k star)，快速入門