卷積神經網路原理及其C++/Opencv實作(4)—誤反向傳播法

原創 sdff20201029 萌萌噠程序猴 2021-03-22 18:54

轉眼，這一系列的文章我們已經更新到第4篇了，在此列出前面三篇的超鏈接，方便讀者跳轉閱讀：

1.卷積神經網路原理及其C++/Opencv實作(1)
2.卷積神經網路原理及其C++/Opencv實作(2)
3.卷積神經網路原理及其C++/Opencv實作(3)

誤反向傳播的過程，也就是誤差訊息從網路末端的Softmax層向網路起始端的C1層傳播的過程。接續上篇文章的內容，本文我們將從數學公式的角度詳細推導一下5層網路的誤反向傳播的過程。

1. 誤反向傳播的一個簡單例子

以下我們先舉個簡單的例子來說明誤反向傳播的原理與目的。

(1) 最優化模型

假設我們有函數E=f(x)，E是關於x的複合函數：

很明顯，在以上E函數的計算過程中，x為輸入訊號，y3為輸出訊號，t為x的標籤（也即輸入x之後我們期待得到的輸出值）。最後得到的E函數值越小，表示y3與t越接近，也也就是輸出訊號越符合我們的期望。所以我們的**目的就很清楚了：求使E函數取得最小值的參數w1、b1、w2、b2、w3、b3**。

看到這裡，是不是覺得這就是一個典型的最佳化問題？是的，沒錯！把函數E當作目標函數值，為了讓E函數求最小值，可以使用梯度下降法對其參數進行最佳化。

(2) 為什麼要用誤反向誤差傳播法

梯度下降法最最佳的原理，我們在先前的文章已經講過，其關鍵是求得參數的梯度（偏導數），然後沿著梯度的反方向更新參數。

梯度下降法詳解

當目標函數不複雜時，我們通常會使用差分法來求參數的近似梯度值，其中eps為一個較小的值，例如0.1或0.5。

由上式可知，計算一個參數的偏導數需要計算兩次目標函數值。當目標函數很複雜時（例如深度學習網路），本身的計算相當耗時，再加上參數的數量很多，對應的需要計算的偏導數也很多。這樣一來，每次優化參數時在計算偏導數上面就耗費了大量時間，導致優化參數的效率極度低。

基於以上原因，聰明的人們又想出了另一種計算參數梯度的方法：誤反向傳播法：**一層一層地計算梯度，從後向前，使用後一層的梯度資訊來計算前一層的梯度**。這樣一來就避免了透過計算目標函數值來計算梯度。

(3) 複合函數的鍊式求導法則

誤反向傳播法應用了複合函數的求導法則，首先我們來講複合函數的求導原理。我們可以把複合函數理解成不同函數的嵌套，要求複合函數中某一變數的導數時，需要一層一層的求導，然後把各層的求導結果相乘，就是最終的求導結果。例如有一個關於x的複合函數y：

則求y關於x的導數如下：

(4) 誤反向傳播的過程

首先，列出需要求解的梯度如下：

根據鍊式求導法則，首先是求w3、b3的梯度：

其次，求w2、b2的梯度：

最後，求w1、b1的梯度：

觀察上述計算可以知道，鍊式求導是一個累乘的過程，不過當前層的部分累乘因子並不需要全部計算，因為上一層已經算好了：

比如我們記：

其中d3為誤差訊息，它從神經網路的末端傳送到起始段的過程，就是誤反向傳播的過程。如下圖所示：

我們稱d1、d2、d3為局部梯度，也就是說，**每一層的局部梯度就是最終的目標函數對本層的輸入訊號的偏導數**（例如E的輸入訊號是y3，y3的輸入訊號是y2， y2的輸入訊號是y1）。得到這些局部梯度之後，從而得到各個參數的梯度：

得到參數的梯度之後，就可以沿著梯度反方向更新參數啦，其中α為學習率，通常設定適當的經驗值，然後隨著迭代的增加而減少。

2. 5層卷積神經網路的誤反向傳播過程

由上篇文章可知，5層神經網路的最後一層是全連接層，此層又可分為Affine層與Softmax層，因此5層網絡又可細分為6層網絡，再加上衡量輸出訊號是否符合期待的交

叉熵誤差函數，共7層。從起始段到末端依序為：卷積層C1 -->池化層S2 -->卷積層C3 -->池化層S4 --> Affine層--> Softmax層-->交叉熵誤差函數E。

接下來我們來講這7層網路的誤反向傳播過程。如下圖所示：

(1) Softmax層局部梯度的計算

前面文章已經講過，交叉熵誤差函數的表達式為：

其中**t為標籤，Y為Softmax函數的輸出，y即是Affine層的輸出也是本層的輸入**：

此層的局部梯度為：

求E關於Y的偏導數：

接著求Y關於y的偏導數，這裡分為兩種情況：

a. 首先是i=j的情況：

a. 其次是i≠j的情況：

綜合以上兩種情況有：

所以有本層局部梯度的計算：

(2) Affine層局部梯度的計算

Affine層的正向傳播計算式如下式，其中y為Affine層的輸出，x是S4層的輸出訊號也是本層的輸入訊號：

現在，我們來求O5層的局部梯度。首先，為了看清複合函數的層次關係，我們畫出以下關係圖：

由上圖可以看出來，E與10個Y都有關係、每個Y與10個y都有關係、每個y都與所有x都有關係，因此求本層的局部梯度，也就是求E關於x的偏導數時，依複合函數的求導法則可按下式計算，其中$0 \le j < 192$。

(3) 池化層S4局部梯度的計算

池化層的正向傳播是一個向下採樣的過程，也也就是資料量減少了。如果是反向傳播呢，很容易想到，那就是向上採樣的過程了，也也就是恢復向下採樣之前的資料量。例如下圖：

首先，我們要知道，S4層輸出的12個4*4影像，被依序展開成長度為12*4*4=192個數據，然後再輸入O5層。所以反向傳播時，O層輸出的對應的192個局部梯度，又被依序轉換成12個4*4矩陣，然後傳入S4層。

由於池化模式分為均值池化與最大值池化，反向傳播也有對應的兩種傳播方式，以下我們分別闡述原理（假設池化視窗為2*2）。

a. 均值池化的反向傳播

如果是均值池化模式，則將局部梯度除以池化窗口的尺寸2*2=4，然後把除以尺寸的結果填入對應的2*2窗口，如下圖所示：

其中：

5/4=1.25
9/4=2.25
3/4=0.75
6/4=1.5

a. 最大值池化的反向傳播

如果是最大值池化模式，則把局部梯度放到池化之前池化視窗中的最大值位置，然後其它位置填入0，例如池化視窗2*2，池化前，池化視窗中最大值的位置分別為左上、右上、左下、右下，則向上採樣後為：

由上述可知，經過池化層S4的反向傳播之後，12個4*4的局部梯度變成12個8*8的局部梯度，如下式，其中⌊⌋為向下取整符號，如⌊3 /2⌋=⌊1.5⌋=1。

上述是針對每個值的計算，如果寫成矩陣的形式則如下，其中d_{S4}為8*8矩陣，d_A為4*4矩陣。

(4) 卷積層C3局部梯度的計算

前面的文章我們已經講過，C3層輸入的是S2層傳來的6個12*12圖像，其正向傳播的計算式如下，其中"*"為圖像的捲積操作，f為Relu激活函數，Y、y、k均表示二維矩陣。b表示一個偏置值，矩陣加上該偏壓的運算相當於矩陣中每個值都加上該偏置，因此結果還是一個矩陣。同理，矩陣輸入Relu激活函數的操作，相當於矩陣中每個值都輸入激活函數，從而每個值的輸出還是組成一個相同維度的輸出矩陣。

以下是目標函數E與本層的輸入、輸出訊號關係圖，由此可以知道E與所有Y_{C3}有關係、每個Y_{C3}只與對應的一個y_{C3}有關係、每個y_{C3}與所有Y_{S2}都有關係。

根據複合函數的求導法則，本層的局部梯度可按下式計算：

首先我們求Y_{C3}關於y_{C3}的偏導數。我們使用的激活函數為Relu函數，其表達式以及導數表達式如下。其中Y_{C3}和y_{C3}都是8*8矩陣(0≤r<8，0≤c<8)。直觀說，正向傳播時就是矩陣y_{C3}的每個數值輸入Relu函數並輸出一個值，所有輸出值組成相同維度的Y_{C3}矩陣。矩陣y_{C3}的偏導數也類似，對於y_{C3}的每個數值，如果大於0則其導數為1，否則為0，所有導數值同樣組成相同維度的導數矩陣。

接下來是求y_{C3}關於Y_{S2}的偏導數。由於y_{C3}是由Y_{S2}經過卷積運算得到的，反向傳播時也同樣是卷積運算（注意到上述d_{C3}的計算式中有個"*"號，這表示矩陣的捲積運算）。這裡我們只放出結論（具體推導我們再另外深究）：

上式中的rotate180表示將矩陣順時針旋轉180°，例如：

於是有：

上式中的"."表示兩個矩陣的對應位置數值相乘，結果還是相同維度的矩陣，"*"表示兩個矩陣的捲積，由於正向傳播的捲積模式為Valid模式，一張12*12的影像經過卷積之後輸出(12-5+1)*(12-5+1)=8*8的捲積結果影像，那麼為了使8*8影像恢復為12*12影像，反向傳播使用Full卷積模式，輸出(8+5-1)*(8+5-1)=12*12影像。

(5) 池化層S2局部梯度的計算

本層的反向傳播原理與S4一樣，都是池化的逆操作，也即向上取樣。

從C3層傳過來的局部梯度是6個12*12矩陣，經過本層後，得到6個24*24矩陣，如下式，其中d_{C3}為12*12矩陣，d_{S2}為24*24矩陣。

(6) 卷積層C1局部梯度的計算

C1層輸入1個28*28的圖像X，其正向傳播的計算式如下，其中"*"為圖像的捲積操作，f為Relu激活函數，Y、y、k均表示二維矩陣，b表示一個偏置值。

本層與目標函數的層次關係為：

要注意的是，本層是5層網路的起始層，所以誤差訊息反向傳播到這裡就停止了，不需要再往前一層傳播（沒有前一層）。所以本層只要要求E關於y的偏導數即可，不需要求E關於X的偏導數。

上式中，d、Y、y都是矩陣，"."為兩個矩陣中對應位置的值相乘，因此結果是一個相同維度的矩陣。

好了，誤差訊息的反向傳播就傳到這裡，接下來我們來總結一下吧，如下圖所示：

下篇文章我們來講怎麼使用局部梯度更新神經網路的參數，敬請期待！

歡迎掃碼追蹤以下微信公眾號，接下來會不定時更新更加精彩的內容噢～

人工智慧 27 深度學習 26 機器學習 33

人工智慧 目錄

上一篇
卷積神經網路原理及其C++/Opencv實作(3)

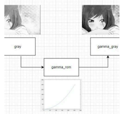
下一篇
卷積神經網路原理及其C++/Opencv實作(5)—參數更新

閱讀原文

喜歡此內容的人還喜歡

數位影像處理之gamma矯正

FPGA開源工作室



混凝土模板荷載與壓力計算

忒修斯破船

NJ系列電子凸輪應用分享

Karl工控