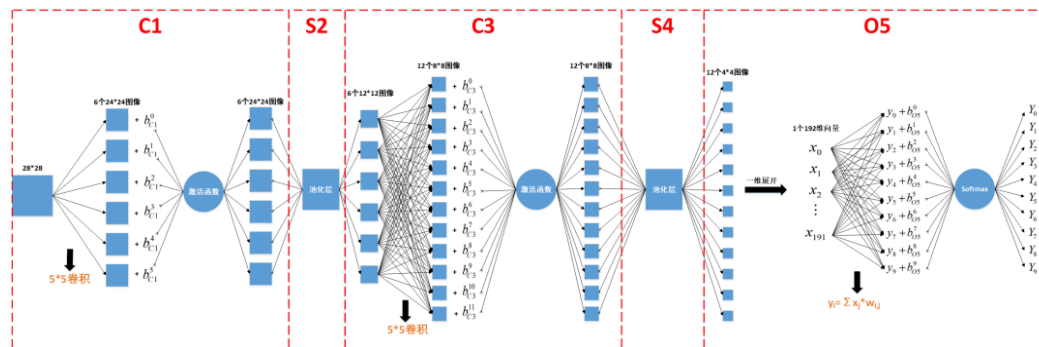


卷積神經網路原理及其C++/Opencv實作(1)

原創 sdff20201029 萌萌噠程序猴 2021-03-08 20:10

2近年來，深度學習大火，在無人駕駛、智慧機器人、圖片辨識及分類、目標偵測、預測未來、疾病診斷等等領域，無一不是深度學習發揮重大、關鍵作用。那麼深度學習到底是什麼，真的有那麼神祕嗎？其實也不然，就像大腦由一個個腦細胞構成一樣，深度學習也是由一個個神經元構成的，如果只看單個細胞或者神經元，覺得沒那麼複雜，但是很多很多細胞或者神經元組合起來形成一個整體，就變得那麼的神奇與偉大。那麼深度學習與神經網路又有什麼關係呢？很簡單，神經網路的層數加深之後，就變成深度學習神經網路了。

接下來的系列文章中，我們將使用C++和Opencv來實作一個5層的捲積神經網路來對手寫數字圖片進行分類（整體結構如下圖所示）。麻雀雖小五臟俱全，雖然只有5層（2個卷積層C1、C3，2個池化層S2、S4，一個全連接層O5），但基本上涵蓋了深度學習網路的各個組成部分。準備好了嗎？讓我們一步步揭開深度學習與神經網路的神秘面紗吧~



1. 總體認知

我們先不看神經網路裡面的數據處理細節，只看輸入端與輸出端：輸入端是我們輸入神經網路的數據，輸出端是我們想要得到的經過神經網路處理之後的數據。所以神經網路的本質是一種資料處理模型，這個資料處理模型有許多參數，這些參數決定著這個資料處理模型的輸出。如下圖所示：

那麼怎麼樣才能讓輸出資料是自己想要的資料呢，這就牽涉到調參的問題，也也就是調整神經網路裡面的參數，這個調整參數的過程我們稱為學習或訓練，神經網路正常運作之前必須經過學習訓練。因此準備並製作好訓練資料非常關鍵。訓練資料包含輸入資料和標籤，標籤就是我們對輸入資料的期待輸出數據，例如輸入1，我們期待輸出2，那麼(1, 2)就組成了一個訓練資料。

訓練的大概過程如下圖所示，具體怎麼使用輸出資料與標籤來調節參數，我們在後續內容再詳細說明。

2. 神經元的概念

這裡說的神經元是電腦科學領域的神經元，不過它是人們由生物科學領域的神經元啟發而提出來的。不得不感慨大自然萬物的奇妙之處，我們人類需要向自然萬物學習的東西還有很多很多~

如下圖所示，一個神經元包括輸入訊號 x_1 、 x_2 ，權重 w_1 、 w_2 ，偏執 b ，活化函數 $f(x)$ 、輸出訊號 y 。當然，實際輸入訊號可能不只2個，也可能有1個、3個或多個，對應的，權重也是可能有1個或更多。

通常在示意圖中，人們喜歡把乘法、加法、偏移、活化函數合成一部分，如下圖：

根據上圖，神經元的輸出 y 依照下式計算：

如果有多個輸入訊號，那麼按照下式計算：

神經網路由多個神經元級聯而成，前一個神經元的輸出作為後一個神經元的輸入，例如下圖：

3. 激活函數

最常見的激活函數有Sigmoid函數和Relu函數。這裡我們作為入門，只講這兩種激活函數，其它的激活函數後續再逐一來。

(1) Sigmoid函數的表達式和函數曲線如下：

(2) Relu函數的表達式和函數曲線如下：

4. 卷積神經元

卷積神經元與神經元類似，主要差異在於神經元是輸入訊號乘以權重，而卷積神經元則是輸入訊號與卷積核（相當於權重）進行卷積。

首先我們來講一下卷積的操作，其實原理很簡單。假設有一張 m 行 n 列的圖像，以及一個 r 行 c 列的捲積核（ **r 和 c 都是奇數，且 $m > r$ ， $n > c$** ）。舉個例子，取 $r=c=3$ ，那麼對於影像中任一點 A ，其捲積值為：在影像上取以點 A 為中心的 $3*3$ 窗口，得到9個點的像素值，與 $3*3$ 捲積核中9個點的值進行對應位置的相乘，最後把9個乘積的結果累積起來，就是捲積值。如上圖所示， x_1 與 k_1 位置對應、 x_2 與 k_2 位置對應、...、 x_9 與 k_9 位置對應，則任意點 A 的捲積值依下式計算：

卷積的過程相當於按照由左往右、從上往下的順序計算原始影像中像素點的捲積值，將計算所得的所有捲積值依照行列排序組成一個二維矩陣，也就是捲積結果。這裡有一個步長 $step$ 的說法：

(1)若步長為1，則由左往右、從上往下地計算原始影像所有點的捲積值。

(2)若步長為2，則由左往右、從上往下地計算原始影像中間隔為1的點的捲積值。

(3)若步長為3，則由左往右、從上往下地計算原始影像中間隔為2的點的捲積值。

如果步長為4、5、6也類似。**不過我們通常預設步長為1，本文後續內容以及後續文章中也皆預設步長為1。**

看到這裡也許有人會有疑問了，如果圖像上每個點都取以它為中心的矩形窗口，那麼邊緣點怎麼辦呢，根本取不到完整的窗口呀？這就需要邊緣填充0值了，也稱為padding操作。根據padding的行列數不同，可以把捲積模式分為**Full**、**Same**、**Valid**三種模式，以下分別說明。

(1) Full模式

在Full模式下，輸出 $m+r-1$ 行 $n+c-1$ 列的捲積結果。

影像的頂部、底部分別需要填滿的行數為：

$$p_r = r - 1$$

影像的左側、右側分別需要填滿的列數為：

$$p_c = c - 1$$

填滿如下圖所示，黑色虛線框內區域為原始影像，藍色區域為邊緣填滿之後的影像，紅色虛線框內的所有點都需要計算卷積值，因此最後輸出 $(m+r-1) * (n+c-1)$ 的捲積結果。

(2) Same模式

在Same模式下，輸出 m 行 n 列的捲積結果。

影像的頂部、底部分別需要填滿的行數為：

$$p_r = (r-1)/2$$

影像的左側、右側分別需要填滿的列數為：

$$p_c = (c-1)/2$$

填滿如下圖所示，紅色虛線框內區域為原始影像，藍色區域為邊緣填滿之後的影像，紅色虛線框內的所有點都需要計算卷積值，因此最後輸出 $m*n$ 的捲積結果。

(3) Valid模式

在Valid模式下，輸出 $(m-r+1)$ 行 $(n-c+1)$ 列的捲積結果。在這個模式下，不需要進行padding操作，直接在原始影像計算卷積，如下圖所示，紅色框內區域為原始影像，灰色區域中的所有點均要計算卷積值，因此最後輸出 $(m-r+1)*(n-c+1)$ 的捲積結果。

好了，講完二維影像的捲積原理，接著我們來講卷積神經元吧。卷積神經元與上文所述的神經元類似，只是乘以權重的操作變成捲積操作了。如下圖所示，其中：

- (1) 兩個卷積結果的相加操作，也即矩陣中對應位置值的相加。
- (2) 加上偏執的操作，也即矩陣中每個值都加上相同的偏執 b 。
- (3) 透過啟動函數的操作，也即矩陣中每個值都輸入激活函數，然後所有的激活函數輸出值組成相同維度的矩陣，該矩陣就是卷積神經元的輸出。

與神經網路相似，卷積神經網路也由多個卷積神經元級聯而成，前一個卷積神經元的輸出作為後一個卷積神經元的輸入。同樣，卷積神經元的輸入不只如上圖有兩張影像，也可能有1、3、4、5.....張影像。

本文就講到這裡啦，下一篇文章我們繼續哈，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～

影像處理 32 機器學習 33 人工智慧 27 深度學習 26

影像處理: 目錄

上一篇

影像配準系列之「Sift特徵點+薄板樣條變換+FFD變換」配準方法

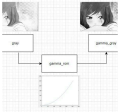
下一篇

基於「FFD形變+梯度下降優化」影像配準的加速方法

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



