



捲積 神經網路介紹

Sean , 王家祥

Table of Contents

1. 捲積神經網路介紹

簡介

2. 捲積

捲積核 -kernel
捲積運算

$$a^2 + b^2 = c^2$$

3. 池化

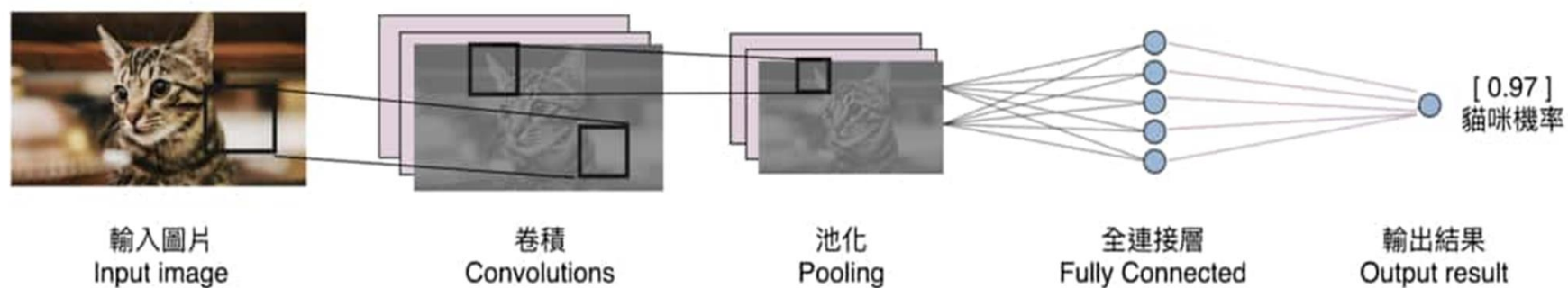
池化-過程
池化--效果

4. Alexnet 模型

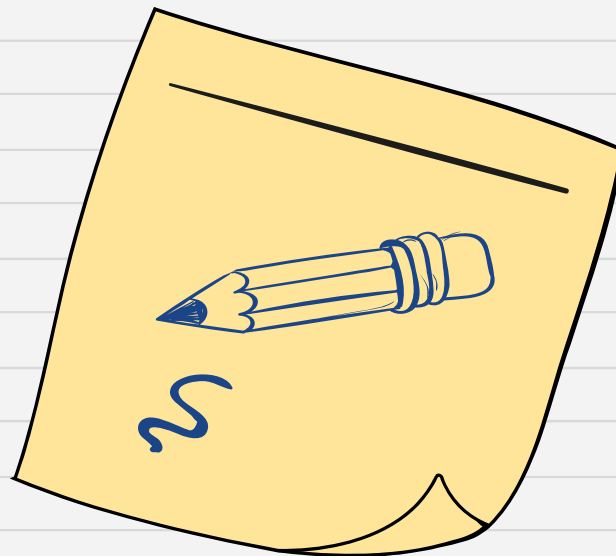
介紹
Demo

1. 捲積神經網路介紹

- 卷積神經網路 (Convolutional Neural Network, 後簡稱為 CNN) 是一種神經網路架構，近年在人臉辨識、圖像分類、自動駕駛等領域大放異彩。本篇供你互動的貓狗辨識應用便是基於此所建立的。



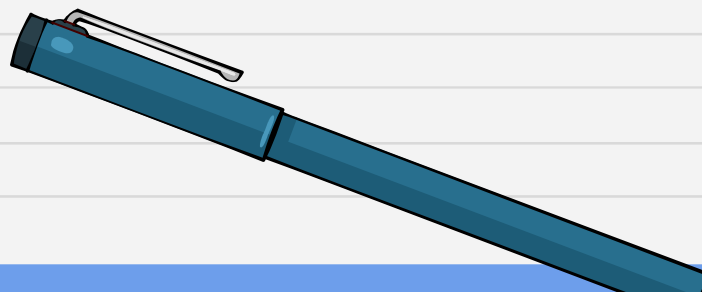
捲積



捲積核 - kernel

捲積運算

捲積運算 -edge_dection-code



2.1 捲積核 -kernel

- 定義: 提取特徵的矩陣, 一般為2維矩陣, 又稱filter

0	-1	0
-1	5	-1
0	-1	0

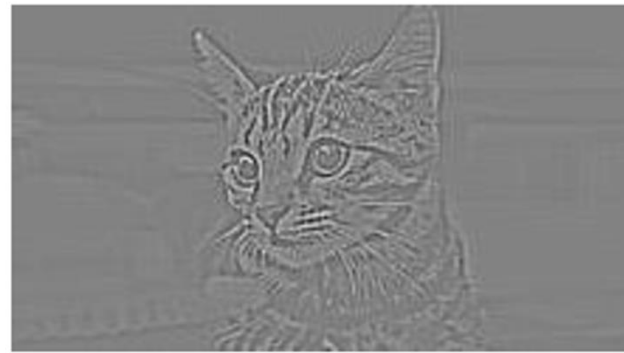
銳利 Sharpen



2.1 捲積核 -kernel

-1	-1	-1
-1	8	-1
-1	-1	-1

輪廓 Outline



2.1 捲積核 -kernel

0.625	0.125	0.625
0.125	0.25	0.125
0.625	0.125	0.625

朦朧 Blur



2.1 捲積核 -kernel

- 除了視覺上的娛樂效果以外，每個濾波器對輸入圖片做的卷積運算事實上是一種[特徵提取（feature extraction）](<https://zh.wikipedia.org/wiki/特徵提取>)步驟。
- 不同的濾波器會對圖中所有相同範圍（3 x 3）的像素做不同的轉換，進而從這些像素裡頭提取出：
 - - 物件輪廓
 - - 左上突出
 - - [邊緣線條](<https://zh.wikipedia.org/wiki/邊緣檢測>)

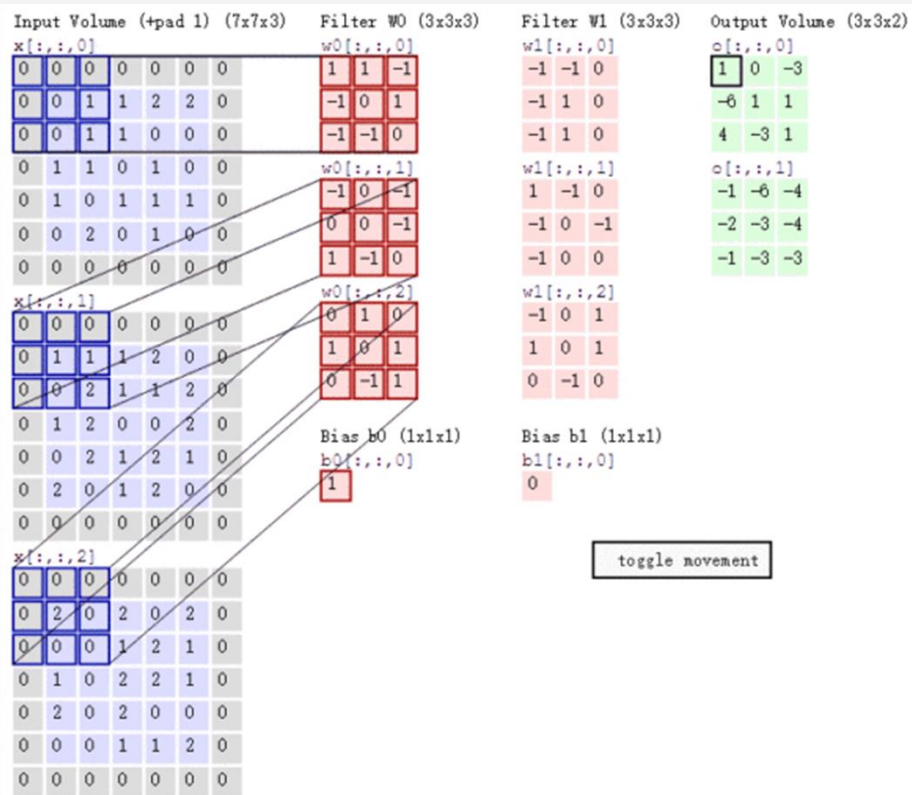
2.2 卷積運算

- 卷積就是將濾波器裡頭的每個數字拿去跟圖片對應位置的像素值相乘，再把所有相乘結果加起來

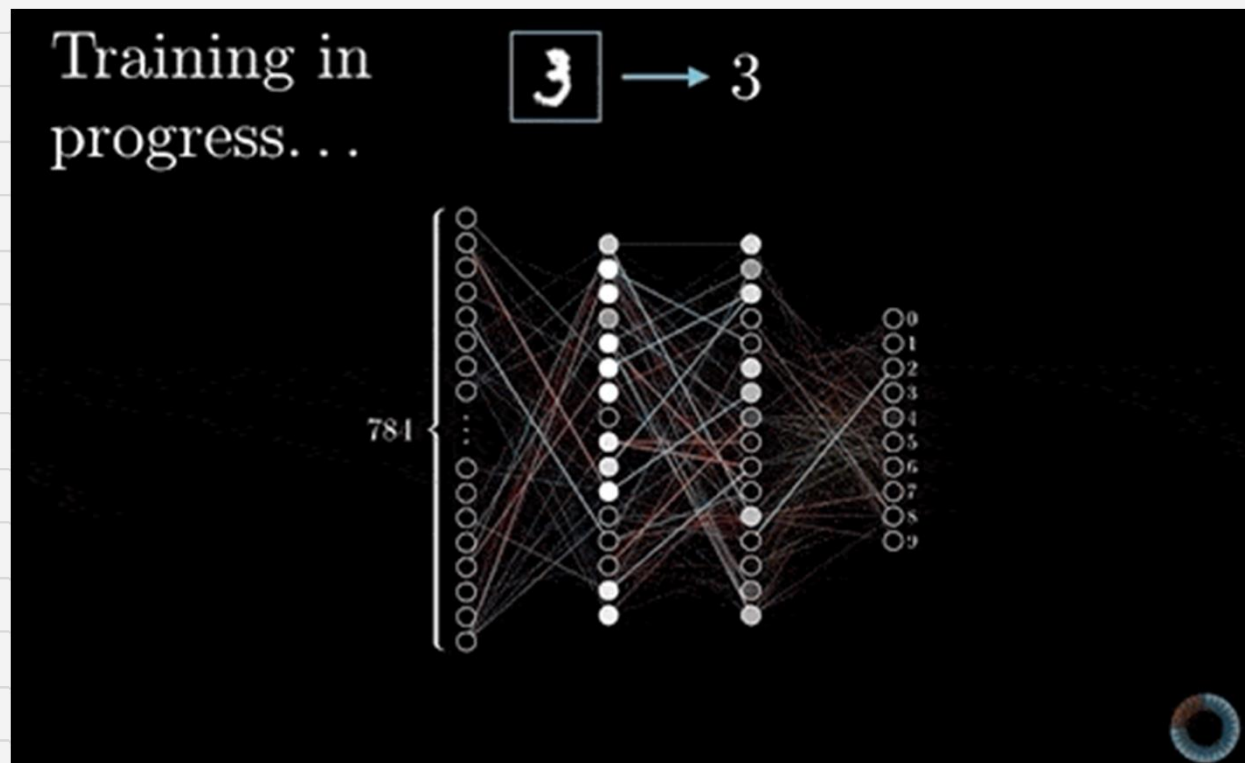
輸入		核		輸出																	
<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8	*	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr></table>	0	1	2	3	=	<table><tr><td>19</td><td>25</td></tr><tr><td>37</td><td>43</td></tr></table>	19	25	37	43
0	1	2																			
3	4	5																			
6	7	8																			
0	1																				
2	3																				
19	25																				
37	43																				

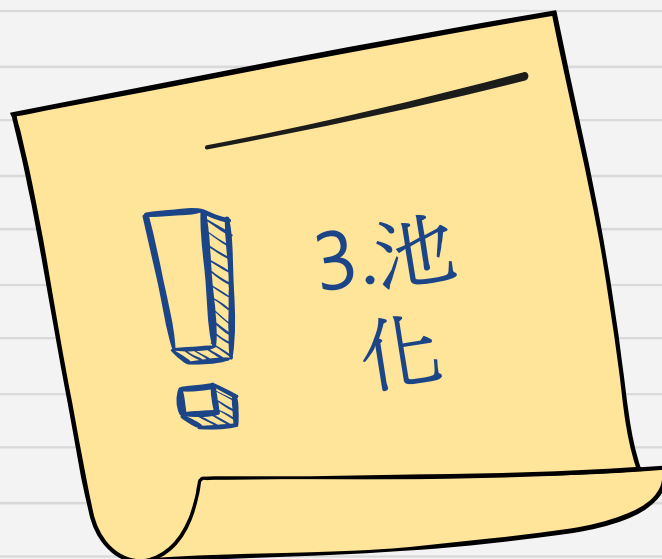
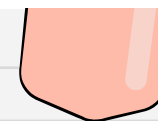
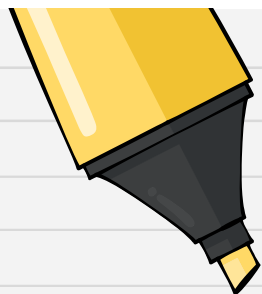
$$\begin{aligned}0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 &= 19, \\1 \times 0 + 2 \times 1 + 4 \times 2 + 5 \times 3 &= 25, \\3 \times 0 + 4 \times 1 + 6 \times 2 + 7 \times 3 &= 37, \\4 \times 0 + 5 \times 1 + 7 \times 2 + 8 \times 3 &= 43.\end{aligned}$$

2.2 捲積運算



2.2 捲積運算-係數更新

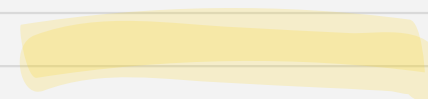




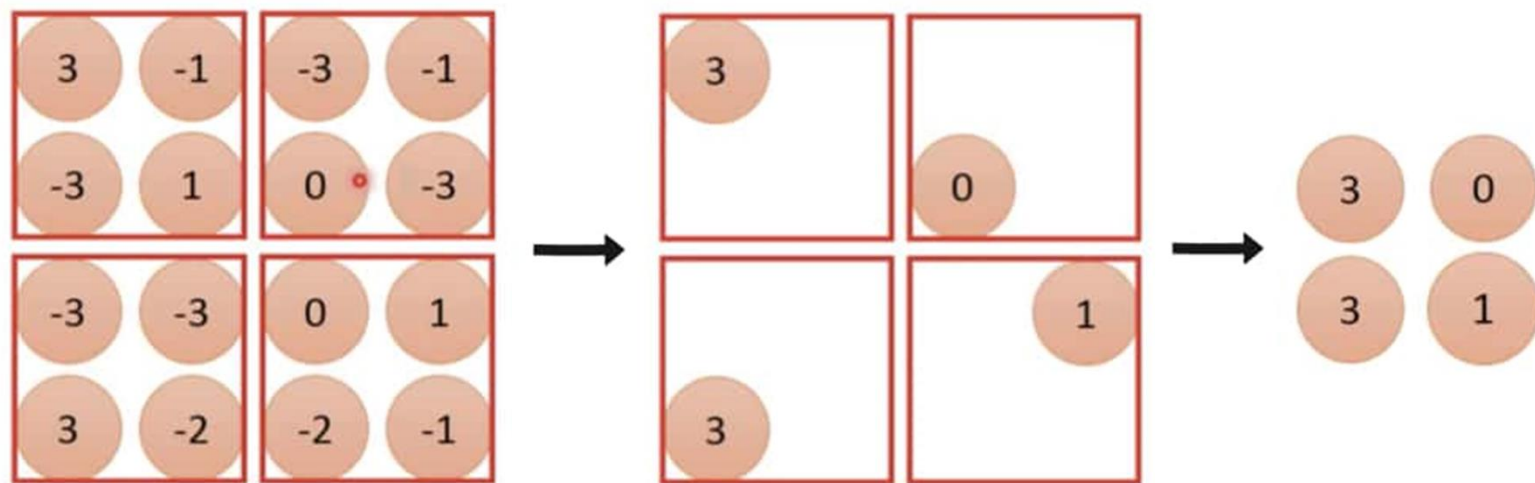
目的:降低取樣, 減低運算複雜度

Average-pooling: 取均值

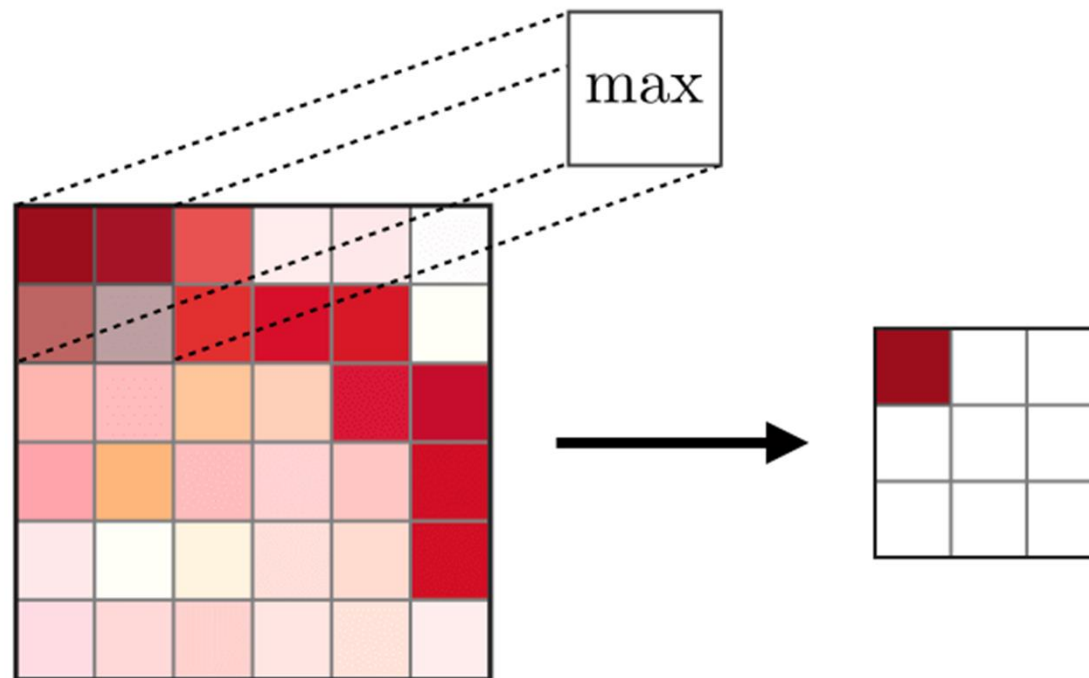
Max-pooling: 取最大值



3.池化-Max pooling



3.池化-過程



3.池化--效果

直觀來說，一個 2×2 的池化運算讓我們在不影響物體的情況下，將原圖大小縮減到原來的 25 %。

另外池化後能避免特徵因為改變位置而沒有被捕獲到



降低取樣
downsampling

3.池化 -Example

http://www.gunniliang.com/notebooks/Delete/git_r/two_month_report/202011_2021_1/11_9_to_11_13_second/code/pooling_ex.ipynb

```
def pool2d(X, pool_size, mode="max"):
    p_h, p_w = pool_size
    Y = tf.zeros((X.shape[0] - p_h + 1, X.shape[1] - p_w + 1))
    Y = tf.Variable(Y)
    for i in range(Y.shape[0]):
        for j in range(Y.shape[1]):
            if mode == "max":
                Y[i,j].assign(tf.reduce_max(X[i:i+p_h, j:j+p_w]))
            elif mode == "avg":
                Y[i,j].assign(tf.reduce_mean(X[i:i+p_h, j:j+p_w]))
    return Y
```

捲積網路概略整理:

- CNN 是一個利用卷積與池化對輸入圖片做特徵擷取的神經網路架構

- 對圖片做卷積是有意義的數據轉換是因為：



- 很多圖形 pattern 尺寸比原圖小很多
- 同樣 pattern 會重複出現在很多地方

- 對圖片做池化是有意義的數據轉換是因為：



- 對像素降低取樣並不會改變圖中物件
- 減少神經網路所需處理的數據量

- 卷積後常跟著池化運算，而你可以重複做（卷積 -> 池化）步驟多次來萃取圖片特徵

- 最後得到的圖片特徵可以交給 CNN 裡的全連接層，由它為我們做（ex:貓狗）分類

- CNN 裡的卷積、池化扮演著萃取特徵的角色，而全連接層則扮演著分類器的角色

AlexNet

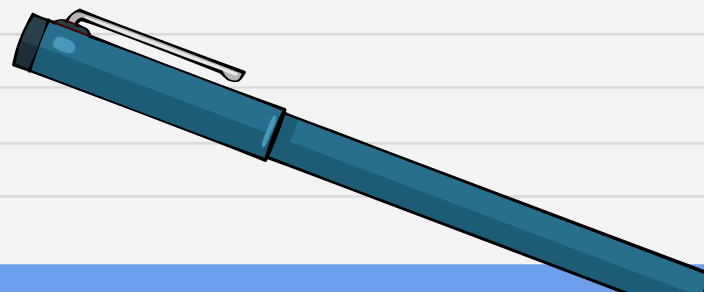
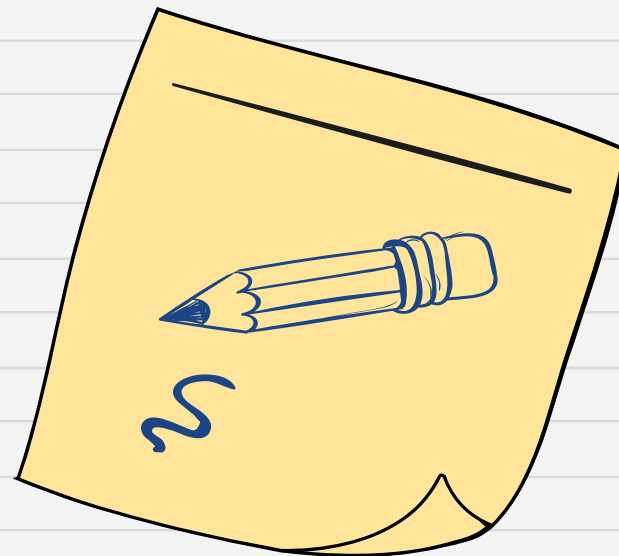
Alexnet 模型

激活函數

降低 Overfitting 的方法

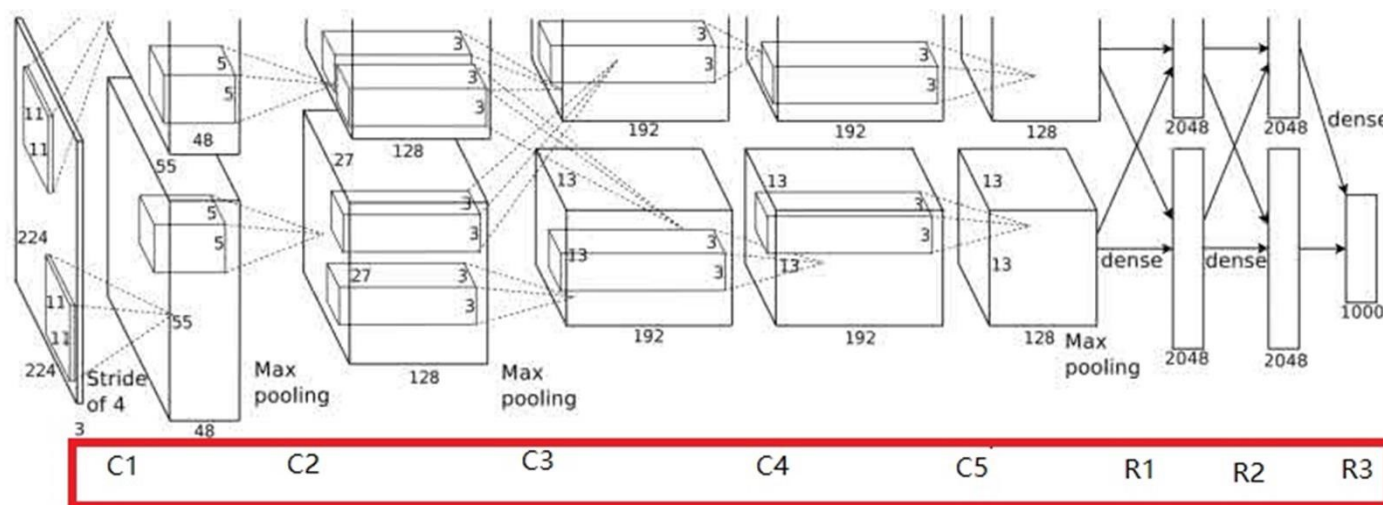
其他

代碼



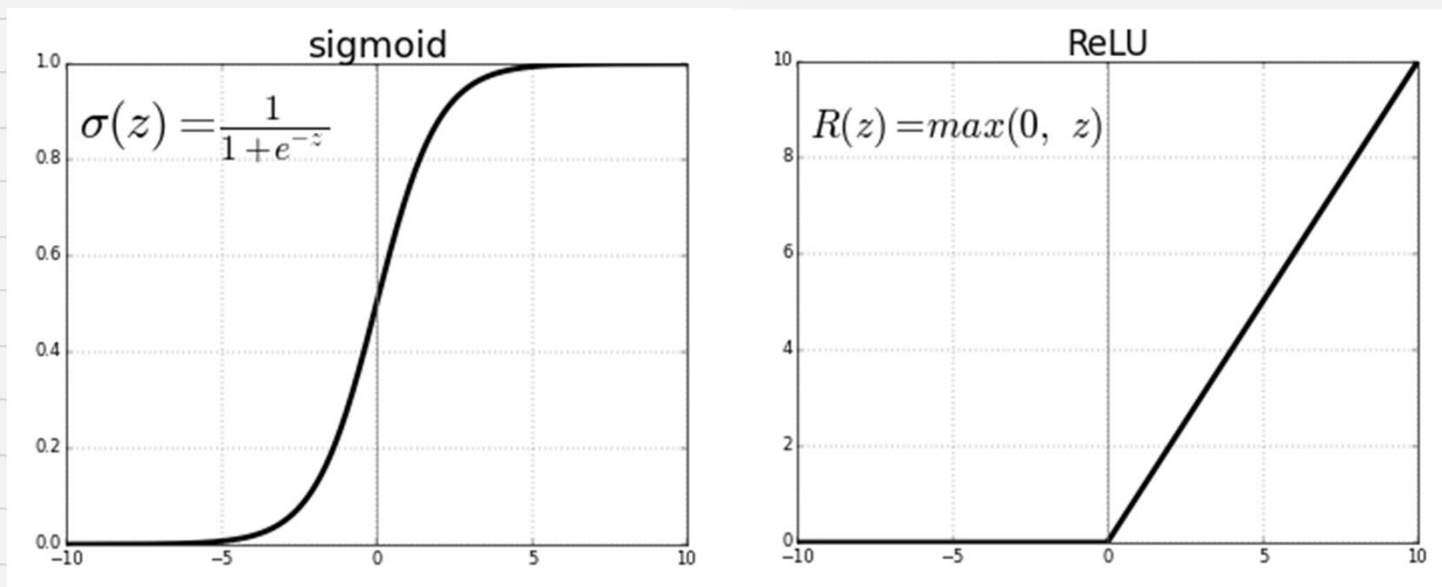
Alexnet 模型

- 2012年，AlexNet橫空出世。這個模型的名字來源於論文第一作者的姓名Alex Krizhevsky。AlexNet使用了8層卷積神經網絡，並以很大的優勢贏得了ImageNet 2012圖像識別挑戰賽。
- 它首次證明了學習到的特征可以超越手工設計的特征，從而一舉打破計算機視覺研究的前狀。
- AlexNet 的架構共八層，第一層到第五層是 Convolutional Layers 做卷積操作（Convolution Operation）與池化（Pooling）；第六層到第八層是 Fully Connected Layers。



4.Alexnet 模型-激活函數

- Activation Function：使用 ReLU 當時最常用的激活函數（Activation Function）是 Sigmoid 和 tanh function。Sigmoid 是把輸入的值壓縮到 0 和 1 之間，但它有`梯度消失`的缺點會讓神經網絡很難被優化



4.Alexnet 模型-降低 Overfitting 的方法

- Dropout 在 AlexNet 中，第六層和第七層的全連階層使用 Dropout，配置為 0.5，表示每個神經元有 50% 的機率不參與下一層的傳遞。這種技術迫使神經網絡需要學習更為穩健的特徵，因此可有效降低 Overfitting。
- Data augmentation AlexNet 的處理分為兩種方式：
 - 原始圖片的像素是 $256 * 256$ ，進行隨機抽取其中的 $224 * 224$ ，且允許水平翻轉將資料擴增 2048 倍。論文中說明此作法能有效的避免 Overfitting。
 - 改變RGB通道的強度：對 RGB 色彩空間做主成份分析（PCA），接著用高斯隨機擾動。這個方法是透過自然圖片的性質來實現，也就是該物體對於照明的強度和顏色的變化是不變的。透過這個方法，top-1 的錯誤率下降 1%。

4.Alexnet 模型-其他特點

- 大數據訓練

百萬數量級的 ImageNet 圖像數據所訓練出來的模型。

- GPU實現

AlexNet 在訓練時用了兩塊 GTX 580 GPU，由於內存 3GB 仍不足以處理如此大量的資料，因此神經網絡分成上下兩層，一塊 GPU 處理一部分的數據，並在中間部分層做溝通。

Code Demo:

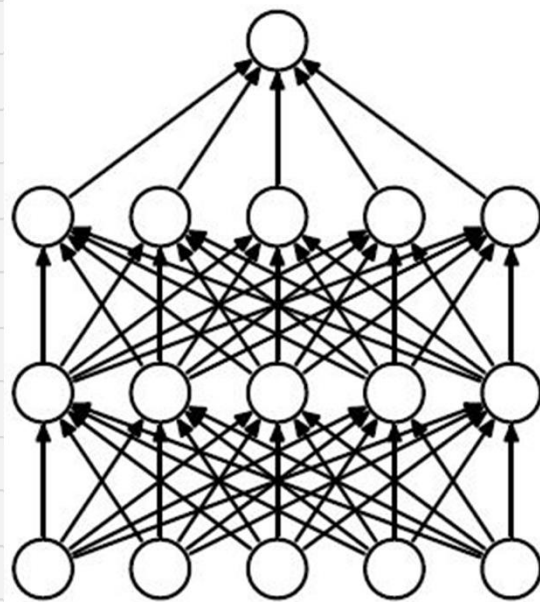
補充 Dropout, Batch Normalization, Adam 概念

- http://www.gunniliang.com/notebooks/Delete/git_r/two_month_report/202011_2021_1/11_9_to_11_13_second/code/Simple_Alexnet.ipynb
- http://www.gunniliang.com/notebooks/Delete/git_r/two_month_report/202011_2021_1/11_9_to_11_13_second/code/AlexNet_Ex.ipynb

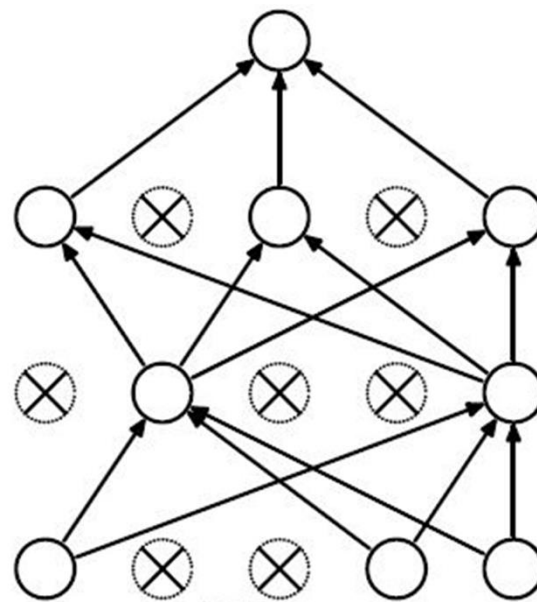
補充 Dropout

- 在机器学习的模型中，如果模型的参数太多，而训练样本又太少，训练出来的模型很容易产生过拟合的现象。
- 在训练神经网络的时候经常会遇到过拟合的问题，过拟合具体表现在：模型在训练数据上损失函数较小，预测准确率较高；但是在测试数据上损失函数比较大，预测准确率较低。
- 總結原因：
 - (1) 容易过拟合
 - (2) 费时
- Dropout可以比较有效的缓解过拟合的发生，在一定程度上达到正则化的效果。

補充 Dropout



(a) Standard Neural Net

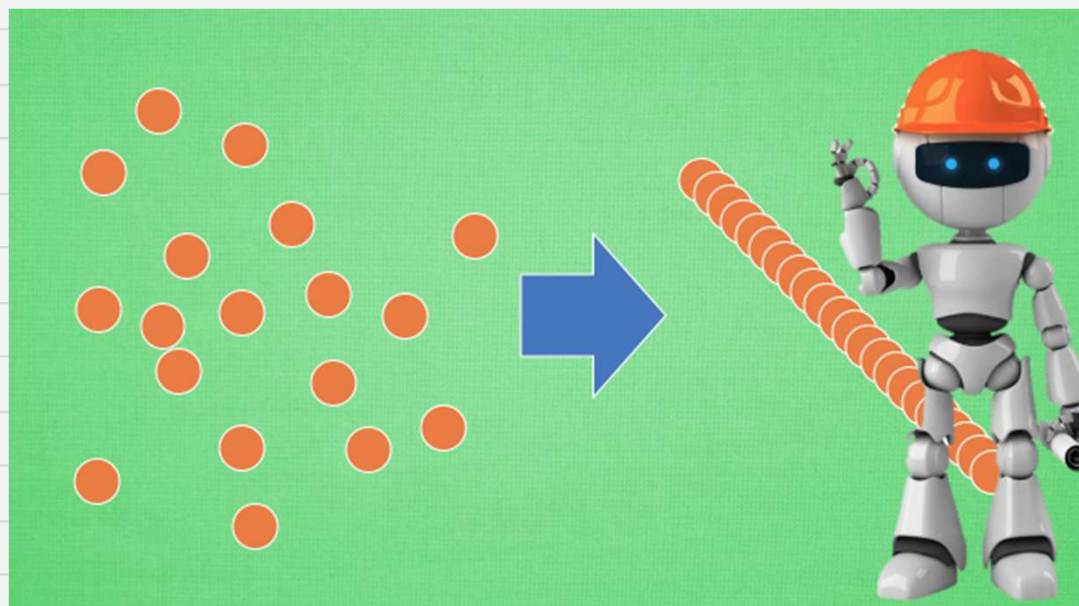


(b) After applying dropout.

知乎 @Microstrong

補充 Batch Normalization

- Batch Normalization, 批标准化, 和普通的数据标准化类似, 是将分散的数据统一的一种做法, 也是优化神经网络的一种方法. 在之前 Normalization 的简介视频中我们一提到, 具有统一规格的数据, 能让机器学习更容易学习到数据之中的规律



補充 Adam



Thanks!



Do you have any questions?

Sean

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), and infographics & images by [Freepik](#).

捲積神經網路介紹

