Convolutional Neural Networks

Page No.
Date

Week 1

Kernel
$$\begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix}$$
→ vertical edge detection
↷ Horizontal

light ↔ dark

Sobel filter
$$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

→ $n \times n$ $*$ $f \times f$ $\longrightarrow$ $(n-f+1 \times n-f+1)$

Image ↑ Convo Kernel

Padding (P) $(n+2p-f+1 \times n+2p-f+1)$

Input = Output → $n = n+2p-f+1$

$p = \frac{f-1}{2}$

→ Strided Convolution (Jump)

Output = $\frac{n+2p-f}{s} + 1$

→ Volume,

$n \times n \times n_c$ $*$ ff nè $\xrightarrow{\text{s}}$ $(n-f+1 \times n-f+1) \times$ nè filters

→ $z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = g[z^{1}]$

→ Input : $n_H^{[l-1]} \times n_W^{[l-1]} \times n_c^{[l-1]}$

Output : $n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$

$$n_H^{[l]} = \left[\frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right]$$

$A^{[l]} \rightarrow m \times n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$

For parameters always check filters

Page No.
Date

Each filter : $f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$

Activations: $a^{[l]} \rightarrow n_H^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

Weights : $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$

bias $n_c^{[l]}$

→ Convolution → Pool → Fully connected

max avg

WEEK 2

→ Residual Network

Direct add (skip)

→ Network in network or 1×1 convolution,
basically you multiply and it behaves like neural network
or also shrink the volume

→ Inception model → Does all type of computation, reduces cost by 1/10

→ MobileNet

→ Depthwise Separable

= Depthwise × Pointwise

Cost summary $= \frac{1}{n_c'} + \frac{1}{f^2}$

Week 3

Sliding windows detection

YOLO - You only look once, we convole the image into a final $3 \times 3 \times 8$ or $19 \times 19 \times 8$ tensor, and the 8 is the output features like $[0/1 \; b_w \; b_h \ldots\ldots]$

Intersection over union $\geq 0.5 \Rightarrow$ correct

Achor box → For 2 objects

$$\begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Transpose convolution

Semantic segmentation.

Low resolution + high info and high res + low info

Week 4

→ Verification
↳ Recognition

→ One shot learning

→ Siamese network
Parameters of NN define an encoding $f(x^{(i)})$

If x^i x^j are same person $\|f(x_i) - f(x_j)\|^2$ is small

$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha \leq 0$ (merging)

→ 3 images

$$L(A, P, N) = \max\left(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0 \right)$$

$$J = \sum_{i=1}^{m} L\left(A^{(i)}, P^{(i)}, N^{(i)}\right)$$

$$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_k \left| f(x^{(i)})_k - f(x^{(j)})_k \right| + b \right)$$

→ Neural style transfer

Select that pixel that maximises the activation function.