LISTAS CNN - DEEP LEARNING

THIAGO MALTA COUTINHO - 2014123335

LISTA I

① a) $300 \times 300 \times 3 \rightarrow 270.000$ entradas

100 neurônios $\Rightarrow (270.000 \times 100) + 100 = 2.700.100$

b) Filtro $5 \times 5 \times 3$ $\quad [(5 \times 5 \times 3) + 1]\, 100 = 7.600$ parâmetros

↳ 100 FILTROS

② Irá detectar bordas verticais

③ a) $f = \begin{bmatrix} -4 & 0 & 4 \\ -4 & 0 & 4 \\ -4 & 0 & 4 \end{bmatrix}$

b) $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 3 \\ -1 & 0 & 1 \end{bmatrix}$

④ $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & -1 & 1 \\ -2 & -2 & 0 & -2 & -2 \\ -1 & -4 & -3 & -4 & -1 \\ -1 & -2 & -1 & -2 & -1 \\ -2 & -3 & -3 & -3 & -2 \end{bmatrix}$

⑤ a) $8 - 3 + 1 = 6 \times 6$

b) $p = \dfrac{s(n' - 1) + f - n}{2} = \dfrac{7 + 3 - 8}{2} = 1$

padding $= 1$, filtro 8×8

---

⑥ a) $p = 0$ $\quad n' = \dfrac{8-3}{2} + 1 = \lfloor 3{,}5 \rfloor \Rightarrow 3 \times 3$

$s = 2$

b) $p = \frac{2.7 + 3 - 8}{2} = 4{,}5$ $f = 8 \times 8$

⑦ O filtro será aplicado o mesmo número de vezes devido ao same padding, que garante o mesmo volume na saída.

⑧ Convolução 1x1 com apenas um filtro e valid padding.

LISTA II

① a) IN: 63x63x16

$P=0$, $S=2$, $f=7$ → 32 filtros

$m' = \frac{63-7}{2} + 1 = 29$

SAÍDA: 29x29x32

② In: 15x15x8, $p=2$ => 19x19x8

③ $p = \frac{s(m'-1) + f - m}{2}$

$f: 7$

$s: 1$

p: same => $m' = m$

$p = \frac{(63-1) + 7 - 63}{2} = \frac{6}{2} = \boxed{3}$ //

④ a) $m' = 65 - 11 + 1 = 55$

$P=0$

$S=1$

$f = 11 \times 11 \times 3$

In: 65x65x3

$(55 \times 55) \times (11 \times 11 \times 3) = 1.098.075$

b) $m' = \frac{65-11}{3} + 1 = 19$ $(19 \times 19) \times (11 \times 11 \times 3) = 131.043$

c) $(11 \times 11 \times 3)(65 \times 65) = 1.533.675$

d) $(11 \times 11 \times 3)(65 \times 65) = 1.533.675$

(5) a) $m' = \frac{65-5}{1} + 1 = 61$ $(5 \times 5 \times 3)(61 \times 61) = 279.075$

c) $p = \frac{(65-1)+5-65}{2} = 2$ $m' = 65 + 4 - 5 + 1 = 65$

$(65 \times 65)(5 \times 5 \times 3) = 316.875$

(6) $m' = \frac{32-2}{2} + 1 = 16 \Rightarrow 16 \times 16 \times 16$

(7) Im = $6 \times 6 \times 3$

$$\begin{bmatrix} 4 & 9 & 2 & 5 & 8 & 3 \\ 5 & 6 & 2 & 4 & 0 & 3 \\ 2 & 4 & 5 & 4 & 5 & 2 \\ 5 & 6 & 5 & 4 & 7 & 8 \\ 5 & 7 & 7 & 9 & 2 & 1 \\ 5 & 8 & 5 & 3 & 8 & 4 \end{bmatrix}$$

a) $m' = \frac{6-2}{2} + 1 = 3 \Rightarrow 3 \times 3 \times 3$

b) $\begin{bmatrix} 9 & 5 & 8 \\ 6 & 5 & 8 \\ 8 & 9 & 8 \end{bmatrix}$

c) $\begin{bmatrix} 6 & 3{,}25 & 7 \\ 8{,}5 & 9 & 5{,}5 \\ 6{,}25 & 6 & 3{,}75 \end{bmatrix}$

(8) Se for MAX pooling, apenas a entrada do maior valor terá seu peso atualizado. Se for AVG pooling, todas as entradas serão atualizadas.

(9) Porque dado uma matriz de entrada $X \in \mathbb{R}^{m \times m}$ e um filtro $F \in \mathbb{R}^{k \times k}$, apenas $(k)^2$ das m entradas receberão, por vez, as operações do filtro. Então, se um filtro em uma posição é um neurônio, ao percorrer toda a matriz de entrada teremos uma camada com g neurônios, mas cada um conectado à k^2 entradas. Essa combinação não se repete.

(10) IN: 32x32x3

f: 5x5x3 → 8 filtros

p: 0

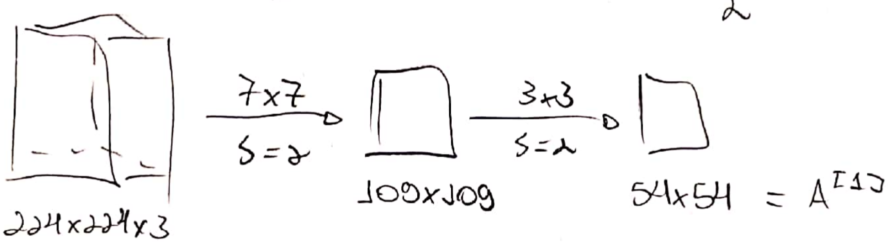
s: 2

$$m' = \frac{32-5}{2} + 1 = \lfloor 14,5 \rfloor$$

OUT: 14x14x8

(11)

$$m' = \frac{224-7}{2} + 1 = 109 \qquad m' = \frac{109-3}{2} + 1 = 54$$



?

(12)

CONV 1 → 608
POOL 1 → 0
CONV 2 → 3216
POOL 2 → 0
FC 3 → 48120
FC 4 → 10164
SOFTMAX → 850

62958

LISTA III

① nH e nW tendem à diminuir e nC tende à aumentar.

② nC → CONV 1x1

nH e nW → CONV 2D com número de filtros = nC

③ In: 28x28x192

f: 5x5 → 32 filtros
p: same
) → OUT: 28x28x32

MULT: $(28 \times 28 \times 32)(5 \times 5 \times 192) = 120.422.400$

f: 1x1 → 16 filtros) → OUT: 28x28x16

MULT: $(1 \times 1 \times 192)(28 \times 28 \times 16) + (5 \times 5 \times 16)(28 \times 28 \times 32) = 12.443.648$

④ O número de operações necessárias ao aplicar os filtros 3x3 e 5x5 será reduzido drasticamente.

⑤ CONV → CONV → POOL → CONV → POOL → FC → FC

⑥ Porque existem problemas como gradient vanishing e gradient explosion em redes profundas sem arquitetura com bloco residual.

⑦ $a^{[l+2]} = g\left(W^{[l+2]} g\left(W^{[l+1]} a^{[l]} + b^{[l+1]}\right) + b^{[l+2]} + a^{[l]}\right)$

⑧ Skip-connections permitem o aprendizado da função identidade e resolvem problemas como vanishing gradient e exploding gradient

Scanned by CamScanner

(9) $(1 \times 16) + 1 = 17$

(10) Podemos reaproveitar as arquiteturas para transfer learning, feature extraction ou modificá-las para gerar novos modelos.

(11) Dois filtros 5x5, introduzir mais não-linearidades à solução. Também utilizamos menos parâmetros.

(12) Pode-se verificar se as camadas mais profundas produzem melhoria significante e com isso, podar a rede caso necessário.