

- I) A rede convolucional LeNET-5 ilustrada na Figura 1 foi desenvolvida por Yann LeCun e publicada em 1989. Sobre a LeNET-5, responda as perguntas abaixo.

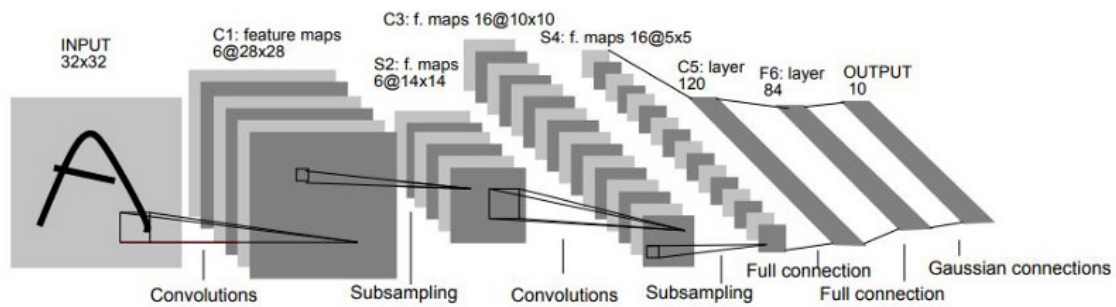


Figura 1: LeNET-5

- a) Qual o motivo do número 5 no nome dessa rede?
 - b) Quais os tamanhos das convoluções usadas?
 - c) Qual o tamanho do campo receptivo de uma unidade dos mapas de ativação 14×14 gerados após a camada de *average pooling* (*subsampling*)?
 - d) Quantos parâmetros treináveis tem essa rede?
 - e) A maior parte dos parâmetros treináveis está localizada nas camadas convolucionais ou nas totalmente conectadas?
- II) O que é o *stride* em uma camada convolucional?
- III) O que é o *dilation* em uma camada convolucional?
- IV) O que é uma convolução transposta?
- V) Por que dizemos que camadas de agrupamento do tipo *Max Pooling* e *Average Pooling* fazem *subsampling* dos mapas de ativação?
- VI) Por que usualmente omitimos o número de canais em uma convolução 2D?
- VII) Considerando uma convolução 2D 3×3 que conecta dois mapas de ativação com tamanho 224×224 com C_1 e C_2 canais respectivamente. Como você poderia inicializar os pesos dessa camada usando a inicialização He?
- VIII) Dada a matriz de input abaixo, e uma rede com 2 camadas convolucionais, calcule a saída da rede com as configurações a seguir. Para a primeira camada convolucional: $\text{Padding} = 0$, $\text{Stride} = 1$; $\text{Kernel} = K_1$; Ativação linear na saída. Para a segunda camada convolucional: $\text{Padding} = 1$; $\text{Stride} = 1$; $\text{Kernel} = K_2$; Ativação ReLU na saída. Considere $\text{dilation} = 0$.

$$I = \begin{bmatrix} 7 & 8 & 9 \\ 6 & 5 & 4 \\ 1 & 2 & 3 \end{bmatrix} \quad K_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad K_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

- IX) Dadas as alternativas abaixo, qual das alternativas representa o Kernel mais indicado para identificação de bordas horizontais em uma imagem?

a)

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

b)

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

c)

$$\begin{bmatrix} 0 & -1 & 0 \\ -2 & 6 & -2 \\ 0 & -1 & 0 \end{bmatrix}$$

d)

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- X) Dada uma imagem de entrada de 32×32 pixels, e duas camadas convolucionais, qual o campo receptivo na imagem original que é representado por um único pixel na saída final? Considere: Kernel para as duas camadas ($K_1 = K_2$) tem tamanho 5×5 ; Stride = 1; Dilation = 2.

Gabarito

- I) a. O número 5 indica o número de camadas com parâmetros treináveis. São 2 convoluções e 3 totalmente conectadas.
- b. Na primeira camada convolucional foram convoluções 5×5 com 6 canais de saída; Na segunda foram convoluções 5×5 com 16 canais de saída.
- c. Tamanho 6×6 . Uma convolução 5×5 tem campo receptivo de tamanho 5×5 , se depois dele fizermos um *average pooling* 2×2 teremos *receptive field* de 6×6 .
- d. Na primeira camada: $5 \times 5 \times 1 \times 6 + 6 = 156$; Na segunda camada: $5 \times 5 \times 6 \times 16 + 16 = 2416$; Na terceira camada $400 \times 120 + 120 = 48120$; Na quarta camada $120 \times 84 + 84 = 10164$; Na quinta camada $84 \times 10 + 10 = 850$. Total 61706.
- e. Nas camadas totalmente conectadas.
- II) É o quanto o filtro translada entre cada uma das aplicações. Por padrão em convoluções usamos 1, o que significa que o kernel é translado de uma posição após cada aplicação do filtro.
- III) É um parâmetro usado para aumentar o campo receptivo da convolução. Na prática tornamos o kernel esparsos completando com zeros.
- IV) É uma convolução usada para fazer *upsampling* na imagem. É uma forma de aprender a função de *upsampling*.
- V) Pois normalmente usamos essas camadas para diminuir a resolução espacial dos mapas de ativação.
- VI) Pois usualmente o número de canais da convolução é igual ao número de canais dos mapas de ativação.
- VII) A inicialização He leva em conta o número de parâmetros que contribuem para cada ativação. Como nesse caso estamos falando de $C_1 \times 3 \times 3$ pesos, podemos inicializar com uma normal com média zero e variância $\frac{2}{9C_1}$.
- VIII) Resolução Input 3×3 Matrizes Iniciais

$$I = \begin{bmatrix} 7 & 8 & 9 \\ 6 & 5 & 4 \\ 1 & 2 & 3 \end{bmatrix} \quad K_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad K_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Camada 1 (Conv1) Input 3×3 , Kernel 3×3 , $p = 0$, $s = 1 \Rightarrow$ Saída 1×1 .

$$C_1 = \sum (I \cdot K_1) + B_1 \quad (B_1 = 0)$$

$$C_1 = (7 \cdot 1 + 8 \cdot 0 + 9 \cdot -1) + (6 \cdot 1 + 5 \cdot 0 + 4 \cdot -1) + (1 \cdot 1 + 2 \cdot 0 + 3 \cdot -1)$$

$$C_1 = (-2) + (2) + (-2) = -2$$

$$C_1 = [-2]$$

Camada 2 (Conv2) Input 1×1 , Kernel 3×3 , $p = 1$, $s = 1 \Rightarrow$ Saída 1×1 . Primeiro, aplicamos padding na saída de C_1 :

$$C_{1,padded} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Em seguida, aplicamos K_2 , $B_2 = 1$ e ReLU:

$$\text{Soma Conv} = \sum (C_{1,padded} \cdot K_2) = (-2 \cdot 0) = 0$$

$$\text{Pré ativação} = \text{Soma Conv} + B_2 = 0 + 1 = 1$$

$$\text{Output} = \text{ReLU}(1) = 1$$

Resultado Final

$$\text{Output} = [1]$$

IX) Alternativa D

- X) A primeira camada de convolução com kernel 5×5 e dilatação 2, resulta em um *feature map* com campo receptivo de 9×9 sobre a imagem de entrada. A segunda camada aplica a mesma configuração de kernel sobre o *feature map* da primeira camada de convolução. Portanto, o campo receptivo final é o campo de visão da camada anterior (9×9) mais a expansão adicional do seu próprio kernel. Resultando em um campo receptivo de 17×17 .