

# CNN - Lista de Exercícios 3

November 9, 2021

## 1 Modelos convolucionais

**1-** Nas camadas mais profundas de uma rede convolucional, comparando com as camadas iniciais, o que tende a ocorrer com as dimensões  $nH$  (altura),  $nW$  (largura) e  $nC$  (número de camadas) das saídas?

**2-** Que tipo de camada você pode usar para reduzir (somente) o  $nC$  de uma entrada? E qual tipo de camada você usaria para reduzir  $nH$  e  $nW$ ?

**3-** Dado um volume de entrada  $28 \times 28 \times 192$  e 32 filtros  $5 \times 5$  com same padding, quantas multiplicações serão executadas? Em uma arquitetura onde são aplicados 16 filtros  $1 \times 1$  antes da convolução com os filtros  $5 \times 5$ , mantendo o restante das operações e dimensões de entrada e saída, quantas multiplicações serão feitas agora?

**4-** Qual é a vantagem de utilizar convoluções de  $1 \times 1$  para reduzir o tamanho do volume de dados de entrada antes de aplicar convoluções de  $3 \times 3$  e  $5 \times 5$  nos blocos de inception?

**5-** Suponha que você irá criar uma rede com 3 camadas convolucionais (CONV), duas camadas de pooling (POOL) e 2 camadas fully connected (FC), em que ordem você colocaria essas camadas na rede?

**6-** Por que a seguinte afirmação é falsa: “O treinamento de uma rede mais profunda (por exemplo, adicionando camadas extras) permite que a rede ajuste funções mais complexas e, portanto, sempre resulta em menor erro de treinamento.”

**7-** Considere um bloco em uma ResNet, expresse a saída desse bloco  $a^{[l+2]}$  em termos de  $W^{[l+2]}$ ,  $W^{[l+1]}$ ,  $b^{[l+1]}$ ,  $b^{[l+2]}$ ,  $a^{[l]}$  e a função de ativação  $g(x)$ .

**8-** Por que as skip-connections (conexões de salto) melhoram a performance da rede? Considere o forward e back-propagation.

**9-** Suponha um volume de entrada  $64 \times 64 \times 16$ . Quantos parâmetros possui um filtro  $1 \times 1$  (incluindo o bias)?

**10-** Quais são motivos mais comuns para o uso de implementações de código aberto de ConvNets (o modelo e/ou pesos)?

**11-** Qual sequência de dois filtros você pode utilizar para atingir o mesmo campo efetivo de um filtro  $9 \times 9$ ? Qual seria a vantagem de fazer essa mudança?

**12-** Na GoogLeNet existe mais de uma camada softmax, como elas melhoram a performance da rede?

## Solução

- 1-  $nH$  e  $nW$  diminuem e  $nC$  aumentam.
- 2- Para reduzir somente o  $nC$ , pode-se utilizar uma camada convolucional  $1 \times 1$ . Já para reduzir o  $nH$  e o  $nW$ , uma camada de pooling pode ser utilizada.
- 3- 120 422 400 e com a convolução  $1 \times 1 : 12\ 443\ 648$ .
- 4- O número de operações necessárias quando utilizado o módulo inception simples é drasticamente reduzido ao se reduzir a dimensionalidade usando convoluções  $1 \times 1$  antes das convoluções  $3 \times 3$  e  $5 \times 5$ .
- 5- Entrada  $\rightarrow$  CONV  $\rightarrow$  CONV  $\rightarrow$  POOL  $\rightarrow$  CONV  $\rightarrow$  POOL  $\rightarrow$  FC  $\rightarrow$  FC  
(ou, alternativamente, duas camadas de CONV antes do POOL final)
- 6- Teoricamente, o erro da rede deveria diminuir ao adicionar mais camadas, na prática, é muito mais difícil para o algoritmo de otimização treinar uma rede profunda. Isso faz com que o erro aumente na rede com mais camadas. Problemas como vanishing e exploding gradients, por exemplo, se tornam mais comuns.
- 7-  $a^{[l+2]} = g(W^{[l+2]}g(W^{[l+1]}a^{[l]} + b^{[l+1]}) + b^{[l+2]} + a^{[l]})$
- 8- As “skip-connections” facilitam que a rede aprenda uma função de identidade entre entrada e saída e ajudam o gradiente a retroceder. Dessa forma, é possível treinar redes mais profundas e evitar problemas de vanishing ou exploding gradients.
- 9- 17
- 10- Se você estiver usando uma arquitetura NN específica que tenha sido treinada anteriormente, poderá usar seus parâmetros/pesos pré-treinados em vez de inicialização aleatória para resolver seu problema (os modelos pré-treinados podem ter sido treinados em grandes conjuntos de dados, como ImageNet, e a reutilização de seus pesos, principalmente nas camadas iniciais pode economizar muito tempo). Além disso, é uma maneira conveniente de trabalhar e implementar arquiteturas mais complexas baseados em redes convolucionais.
- 11- Dois filtros  $5 \times 5$  têm o mesmo campo efetivo do filtro  $9 \times 9$ . Uma vantagem de utilizar mais filtros é introduzir mais não-linearidade (através das funções de ativação) na rede, outra é que são menos parâmetros a serem estimados (50 contra 81).
- 12- Ao “injetar” gradientes adicionais nas camadas intermediárias, a rede garante que as features sendo computadas antes da camada final são relevantes para a classificação da imagem.