

Definindo arquitetura

Ao se definir uma arquitetura, a ordem em que as camadas aparecem é essencial para definir o tipo de processamento que a rede vai realizar. Enquanto as camadas convolucionais transformam o espaço, projetando features cada vez mais semânticas, o pooling auxilia na eficiência da rede e adiciona invariância a pequenas translações. Por fim, as camadas densas (totalmente conectadas) são comumente usadas como classificadores a partir das características projetadas pelas convoluções.

Suponha que você irá criar uma rede com 3 camadas convolucionais (Conv2d), duas camadas de pooling (MaxPool2d) e 2 camadas Fully Connected (Linear). Em que ordem você colocaria essas camadas na rede?

Selecione uma alternativa

A

```
Net = nn.Sequential(  
    nn.Conv2d(3, 16, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.Conv2d(16, 32, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.Conv2d(32, 64, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.MaxPool2d(kernel_size=2),  
    nn.Flatten(),  
    nn.Linear(4096, 1024),  
    nn.ReLU(),  
    nn.MaxPool2d(kernel_size=2),  
    nn.Linear(512, 10)  
)
```

B

```
Net = nn.Sequential(  
    nn.Conv2d(3, 32, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.MaxPool2d(kernel_size=2),  
    nn.Conv2d(32, 64, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.Conv2d(64, 64, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.MaxPool2d(kernel_size=2),  
    nn.Flatten(),  
    nn.Linear(4096, 1024),  
    nn.ReLU(),  
    nn.Linear(1024, 10)  
)
```

C

```
Net = nn.Sequential(  
    nn.Conv2d(3, 16, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.Conv2d(16, 32, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.Conv2d(32, 64, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.MaxPool2d(kernel_size=2),
```

```
nn.MaxPool2d(kernel_size=2),  
nn.Flatten(),  
nn.Linear(4096, 1024),  
nn.ReLU(),  
nn.Linear(1024, 10)  
)
```