

pooling_setup

February 14, 2025

0.0.1 1) A typical setup for Pooling layers

```
[95]: import torch
import torch.nn as nn

layer_with_pooling = nn.Sequential(
    nn.Conv2d(1, 16, 3, 1, 1),
    nn.MaxPool2d(2, 2),
    nn.Conv2d(16, 32, 3, 1, 1),
    nn.MaxPool2d(2, 2)
)
example_input = torch.randn(5, 1, 28, 28)
example_output = layer_with_pooling(example_input)
print(example_output.size())
```

torch.Size([5, 32, 7, 7])

0.0.2 2) without pooling layers

```
[126]: import torch
import torch.nn as nn

layer_with_pooling = nn.Sequential(
    nn.Conv2d(1, 16, 3, 1, 1),
    # nn.MaxPool2d(2, 2),
    nn.Conv2d(16, 32, 3, 1, 1),
    # nn.MaxPool2d(2, 2)
)
example_input = torch.randn(5, 1, 28, 28)
example_output = layer_with_pooling(example_input)
print(example_output.size())
```

torch.Size([5, 32, 28, 28])

0.0.3 3) without pooling layers but with stride=2

```
[131]: import torch
import torch.nn as nn

layer_with_pooling = nn.Sequential(
    nn.Conv2d(1, 16, 3, 2, 1),
    # nn.MaxPool2d(2, 2),
    nn.Conv2d(16, 32, 3, 2, 1),
    # nn.MaxPool2d(2, 2)
)
example_input = torch.randn(5, 1, 28, 28)
example_output = layer_with_pooling(example_input)
print(example_output.size())
```

torch.Size([5, 32, 7, 7])

0.0.4 4) Ví dụ sử dụng tích chập trên ảnh màu trex.png

```
[127]: import torch
import torch.nn as nn
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
from torchvision import transforms

layer_with_pooling = nn.Sequential(
    nn.Conv2d(3, 16, 3, 1, 1), # 3 là số kênh ảnh màu của ảnh đầu vào
    nn.MaxPool2d(2, 2),
    nn.Conv2d(16, 32, 3, 1, 1),
    nn.MaxPool2d(2, 2)
)
# chuyển ảnh màu thành tensor sử dụng transforms từ torchvision
transform = transforms.Compose([
    transforms.Resize((64, 64)),
    transforms.ToTensor()
])
# Đọc ảnh bằng thư viện PIL
img = Image.open("trex.png")
img = transform(img)

# plt.imshow(np.transpose(img.detach().numpy(), (1, 2, 0)))
example_input = torch.stack([img], dim=0)
example_output = layer_with_pooling(example_input)
print("Kích thước feature map khi qua CNN:", example_output.size())
```

Kích thước feature map khi qua CNN: torch.Size([1, 32, 16, 16])

```
[130]: import matplotlib.pyplot as plt

# hiển thị ảnh đầu vào đã transform
plt.figure(figsize=(2, 2))
plt.title('Input: Example 0')
plt.imshow(np.transpose(img.detach().numpy(), (1,2,0)))

# hiển thị feature map của dữ liệu mẫu đầu tiên
plt.figure(figsize=(10, 7))
plt.title('Output: Example 0')
plt.axis('off')

for i, _ in enumerate(example_output):
    for j, _ in enumerate(example_output[i]):
        # vẽ example_output[i][j] trên lưới có 8 cột và 4 hàng (do  $8*4 = 32 = \lfloor$ 
        ↪ số feature map cuối cùng)
        plt.subplot(4, 8, j+1)
        plt.imshow(example_output[i,j].detach().numpy(), cmap='gray')
        plt.axis('off')
        plt.title(f'channel {j+1}')
    break
plt.show()
```



