```
1 from nndl.layers import *
2 from utils.fast_layers import *
3
4
5 def conv_relu_forward(x, w, b, conv_param):
6
7
    A convenience layer that performs a convolution followed by a ReLU.
8
9
    Inputs:
    - x: Input to the convolutional layer
10
11
     - w, b, conv_param: Weights and parameters for the convolutional layer
12
13
     Returns a tuple of:
14
     - out: Output from the ReLU
15
     - cache: Object to give to the backward pass
16
17
     a, conv_cache = conv_forward_fast(x, w, b, conv_param)
18
     out, relu_cache = relu_forward(a)
19
     cache = (conv_cache, relu_cache)
20
     return out, cache
21
22
23 def conv_relu_backward(dout, cache):
24
25
     Backward pass for the conv-relu convenience layer.
26
27
     conv_cache, relu_cache = cache
28
     da = relu backward(dout, relu cache)
29
     dx, dw, db = conv_backward_fast(da, conv_cache)
30
     return dx, dw, db
31
32
33 def conv_relu_pool_forward(x, w, b, conv_param, pool_param):
34
35
     Convenience layer that performs a convolution, a ReLU, and a pool.
36
37
     Inputs:
38
     - x: Input to the convolutional layer
39
     - w, b, conv_param: Weights and parameters for the convolutional layer
40
     - pool_param: Parameters for the pooling layer
41
42
     Returns a tuple of:
43
     - out: Output from the pooling layer
44
     - cache: Object to give to the backward pass
45
46
     a, conv_cache = conv_forward_fast(x, w, b, conv_param)
47
     s, relu_cache = relu_forward(a)
48
     out, pool_cache = max_pool_forward_fast(s, pool_param)
49
     cache = (conv_cache, relu_cache, pool_cache)
50
     return out, cache
51
52
53 def conv_relu_pool_backward(dout, cache):
54
55
     Backward pass for the conv-relu-pool convenience layer
56
57
     conv_cache, relu_cache, pool_cache = cache
58
     ds = max_pool_backward_fast(dout, pool_cache)
59
     da = relu backward(ds, relu cache)
```

localhost:4649/?mode=python 1/2

2022/2/23 清晨7:43 conv\_layer\_utils.py

dx, dw, db = conv\_backward\_fast(da, conv\_cache)
return dx, dw, db

localhost:4649/?mode=python