

```
1 from .layers import *
2
3
4 def affine_relu_forward(x, w, b):
5     """
6     Convenience layer that performs an affine transform followed by a ReLU
7
8     Inputs:
9     - x: Input to the affine layer
10    - w, b: Weights for the affine layer
11
12    Returns a tuple of:
13    - out: Output from the ReLU
14    - cache: Object to give to the backward pass
15    """
16    a, fc_cache = affine_forward(x, w, b)
17    out, relu_cache = relu_forward(a)
18    cache = (fc_cache, relu_cache)
19    return out, cache
20
21
22 def affine_relu_backward(dout, cache):
23     """
24     Backward pass for the affine-relu convenience layer
25     """
26    fc_cache, relu_cache = cache
27    da = relu_backward(dout, relu_cache)
28    dx, dw, db = affine_backward(da, fc_cache)
29    return dx, dw, db
30
```