

Given image x of size $N * N$, convolution kernel w of size $M * M$ and function f . Write python code with no libraries (or pseudocode) that:

a) calculates $y = conv(x, w)$

b) assuming you are also given function $g = \frac{\partial f}{\partial y}$, performs backward pass to calculate $\frac{\partial f}{\partial w}$

Solution:

a)

```
def conv(x, w):
    w_len = len(w)
    y_len = len(x) - w_len + 1

    y = []
    for i in range(y_len):
        row = []
        for j in range(y_len):
            stamp = 0
            for m in range(w_len):
                for n in range(w_len):
                    stamp += w[m][n] * x[i+m][j+n]
            row.append(stamp)
        y.append(row)
    return y
```

```
y = conv(x, w)
```

b)

```
w_grad = conv(x, g(y))
```