Step-by-Step Backpropagation for CNN Filter Update

Step 1: Forward Pass (Convolution)

Let's compute the output feature map by applying the 2×2 filter to the 3×3 input image:

Output(0,0) = I(0,0)×W(0,0) + I(0,1)×W(0,1) + I(1,0)×W(1,0) + I(1,1)×W(1,1)
= 1×0.5 + 2×0.2 + 0×0.3 + 1×0.7 = 1.6

Output(0,1) = I(0,1)×W(0,0) + I(0,2)×W(0,1) + I(1,1)×W(1,0) + I(1,2)×W(1,1)
= 2×0.5 + 1×0.2 + 1×0.3 + 2×0.7 = 2.9

Output(1,0) = I(1,0)×W(0,0) + I(1,1)×W(0,1) + I(2,0)×W(1,0) + I(2,1)×W(1,1)
= 0×0.5 + 1×0.2 + 2×0.3 + 1×0.7 = 1.5

Output(1,1) = I(1,1)×W(0,0) + I(1,2)×W(0,1) + I(2,1)×W(1,0) + I(2,2)×W(1,1)
= 1×0.5 + 2×0.2 + 1×0.3 + 0×0.7 = 1.2

Computed output:

```
[1.6  2.9]
[1.5  1.2]
```

Step 2: Calculate Loss (MSE)

MSE = (1/4) × [(1.6-1)² + (2.9-2)² + (1.5-2)² + (1.2-1)²]
= (1/4) × [0.36 + 0.81 + 0.25 + 0.04]
= (1/4) × 1.46
= 0.365

Step 3: Compute Gradients

First, calculate the gradient of loss with respect to each output element:

∂Loss/∂Output(0,0) = (2/4) × (1.6-1) = 0.3
∂Loss/∂Output(0,1) = (2/4) × (2.9-2) = 0.45
∂Loss/∂Output(1,0) = (2/4) × (1.5-2) = -0.25
∂Loss/∂Output(1,1) = (2/4) × (1.2-1) = 0.1

Next, compute gradients for each filter weight:

∂Loss/∂W(0,0) = ∂Loss/∂Output(0,0) × I(0,0) + ∂Loss/∂Output(0,1) × I(0,1) +
∂Loss/∂Output(1,0) × I(1,0) + ∂Loss/∂Output(1,1) × I(1,1)

$= 0.3 \times 1 + 0.45 \times 2 + (-0.25) \times 0 + 0.1 \times 1 = 1.3$

$\partial Loss/\partial W(0,1) = \partial Loss/\partial Output(0,0) \times I(0,1) + \partial Loss/\partial Output(0,1) \times I(0,2) + \partial Loss/\partial Output(1,0) \times I(1,1) + \partial Loss/\partial Output(1,1) \times I(1,2)$
$= 0.3 \times 2 + 0.45 \times 1 + (-0.25) \times 1 + 0.1 \times 2 = 1.0$

$\partial Loss/\partial W(1,0) = \partial Loss/\partial Output(0,0) \times I(1,0) + \partial Loss/\partial Output(0,1) \times I(1,1) + \partial Loss/\partial Output(1,0) \times I(2,0) + \partial Loss/\partial Output(1,1) \times I(2,1)$
$= 0.3 \times 0 + 0.45 \times 1 + (-0.25) \times 2 + 0.1 \times 1 = 0.05$

$\partial Loss/\partial W(1,1) = \partial Loss/\partial Output(0,0) \times I(1,1) + \partial Loss/\partial Output(0,1) \times I(1,2) + \partial Loss/\partial Output(1,0) \times I(2,1) + \partial Loss/\partial Output(1,1) \times I(2,2)$
$= 0.3 \times 1 + 0.45 \times 2 + (-0.25) \times 1 + 0.1 \times 0 = 0.95$

Step 4: Update Filter Weights

Using a learning rate of 0.1 for gradient descent:

$W(0,0)\_new = 0.5 - 0.1 \times 1.3 = 0.37$
$W(0,1)\_new = 0.2 - 0.1 \times 1.0 = 0.10$
$W(1,0)\_new = 0.3 - 0.1 \times 0.05 = 0.295$
$W(1,1)\_new = 0.7 - 0.1 \times 0.95 = 0.605$

Updated filter:

```
[0.37  0.10]
[0.295 0.605]
```

This updated filter will produce an output closer to the target, reducing the Mean Squared Error loss.

⁂