Since design is modular we just have to think about B-PASS in conv layers.

1. **CNN-B-PASS**
   - This is going to be a notational nightmare
   - So let's simplify thing a bit
   - Say \( C_1 = 1 \) \( C_2 = 1 \) [we can always generalize, it's just an extra sum]
   - And let's not overload \( h \). And let's drop super-script.

   ![Diagram of CNN-B-PASS](image)

   **Input**  \( x \) \hspace{1cm} **Kernel/Filter** \hspace{1cm} **Output** \( y \)

   **Notation:** for iterate \( h \)
Recall: \( y[a,c] = \sum_{a=0}^{k_1-1} \sum_{b=0}^{k_2-1} x[a+a, c+b] w[a,b] \)

\[
\begin{bmatrix}
\vec{w}_1 \\
\vec{w}_2
\end{bmatrix} = \vec{x} \cdot \vec{w}
\]

Note: sizes \( |y| = N_1 \times N_2 \)

Thus, \( \left| \frac{\partial L}{\partial y} \right| = N_1 \times N_2 \)

We will use \( \frac{\partial L}{\partial y[a,c]} \) to access members of incoming gradients.

Need to compute: \( \frac{\partial L}{\partial x} \)  \( \frac{\partial L}{\partial w} \)
\[ \frac{\partial L}{\partial w[a', b']} = \sum_{x=0}^{N_x-1} \sum_{c=0}^{N_c-1} \frac{\partial L}{\partial y[a, c]} \frac{\partial y[a, c]}{\partial w[a', b']} \]

What does this weight affect? Everything! (via \( \frac{\partial L}{\partial w[a', b']} \)).

Visually:

\[ x \rightarrow w[a', b'] \rightarrow y \]

\[ \frac{\partial y[a, c]}{\partial w[a', b']} = x[a + a', c + b'] \]

(can also prove analytically)

\[ \Rightarrow \frac{\partial L}{\partial w[a', b']} = \sum_{x=0}^{N_x-1} \sum_{c=0}^{N_c-1} \frac{\partial L}{\partial y[a, c]} x[a + a', c + b'] \]

Looks like a convolution of 2 \((N_x N_c)\) matrices!

\[ \theta x \ast \frac{\partial L}{\partial y} \]

But clipped to \((k_1, k_2)\)!
Let's do it pixel by pixel.

\( \frac{\partial L}{\partial x} \)

\( \frac{\partial L}{\partial x[i, c]} \)

What does this pixel affect?

[Diagram]

\( x \rightarrow y \)

\( x[i, c] \) affects \( y \) in a neighborhood around \([x, c]\)

Neighborhood size is determined by size of filter \( W \).

In figure above, box 1 represents the "first" box where \( y \) \( x[i, c] \) plays a role.

Box 3 represents the "last" box. Box 2 is in between.

So \( x[i, c] \) affects the following output pixel:

[Diagram]
So let's apply chain rule.

\[ \frac{\partial f}{\partial x[x', c']} = \sum \frac{\partial f}{\partial y[x', c']} \cdot \frac{\partial y[x', c']}{\partial x[x', c']} \]

Let's make this a bit more formal.

\[ \frac{\partial}{\partial x[x', c']} = \sum_{a=0}^{k-1} \frac{\partial}{\partial y[x'-a, c'-b]} \cdot \frac{\partial y[x'-a, c'-b]}{\partial x[x', c']} \]

given \hspace{1cm} \text{Let's compute/derive}

We can 'derive' visually (like last time).

\[ \text{or analytically.} \]

Let's try analytically this time:

\[ y[x', c'] = \sum_{a=0}^{k-1} \sum_{b=0}^{k-1} x[x'+a', c'+b'] \cdot \omega[a', b'] \]

Same equation as a few pages ago, just written with 'primes'.

Now \[ y[x'-a, c'-b] = \sum_{a=0}^{k-1} \sum_{b=0}^{k-1} x[x'-a+a', c'-b+b'] \cdot \omega[a', b'] \]

\[ \frac{\partial y[x'-a, c'-b]}{\partial x[x', c']} = \omega[a, b] \] [why? \[ x[x', c'] \] only appears once in the expression]
\[
\frac{\partial L}{\partial x(x',c')} = \sum_{k=1}^{k=1} \sum_{\alpha=0}^{\beta=0} \frac{\partial L}{\partial y(x'-a, c'-b)} \cdot w[a,b]
\]

Very nice! Almost like a 'convolution' [More like 'cross-correlation']

Actually if we "flip" \( w \) about \( \theta \) it's center horizontally \& vertically to get \( w^{flip} \)

\[
\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \ast w^{flip}
\]

Super cool:

So F-PROP = Convolution

B-PROP = Convolution with flipped filter!