Computing derivatives.

We need to compute $\frac{\partial z}{\partial \varpi_i}$ for each i.

At the end of the network

$$X^{L-1} \longrightarrow \boxed{\varpi^L} \longrightarrow z$$

$$\uparrow \quad \tfrac{1}{2}\| t(X^1) - X^L \|^2$$

it's easy to see

$$\frac{\partial z}{\partial \varpi^L} = 0 \qquad \frac{\partial z}{\partial X^L} = \frac{\partial}{\partial X^L} \frac{1}{2} \left(t(X^1) - X^L \right) \cdot \left(t(X^1) - X^L \right)$$

$$= X^L - t.$$

Now consider

$$X^{L-1} \longrightarrow \boxed{\varpi^{L-1}} \longrightarrow X^L$$

We have (in principle)

$$\frac{\partial z}{\partial \varpi^{L-1}} = \frac{\partial z}{\partial X^L} \cdot \frac{\partial X^L}{\partial \varpi^{L-1}} \qquad \text{(chain rule)}$$

and

$$\frac{\partial z}{\partial X^{L-1}} = \frac{\partial z}{\partial X^{L}} \cdot \frac{\partial X^{L}}{\partial X^{L-1}} \quad \text{(chain rule)}$$

However, we need to keep track of the order (# indices) and range (values for index) of ~~each~~ the tensors X^L, X^{L-1} to make the multiplications above make sense. So we write

$\mathrm{vec}(X^L)$ as a function of $\mathrm{vec}(X^{L-1})$, $\mathrm{vec}(W^{L-1})$

and then

$$\frac{\partial z}{\partial\, \mathrm{vec}(W^{L-1})^T} = \underbrace{\frac{\partial z}{\partial\, \mathrm{vec}(X^L)^T}}_{\text{vectors}} \; \underbrace{\frac{\partial\, \mathrm{vec}(X^L)}{\partial\, \mathrm{vec}(W^{L-1})^T}}_{\text{matrices}}$$

$$\frac{\partial z}{\partial\, \mathrm{vec}(X^{L-1})^T} = \frac{\partial z}{\partial\, \mathrm{vec}(X^L)^T} \; \frac{\partial\, \mathrm{vec}(X^L)}{\partial\, \mathrm{vec}(X^{L-1})^T}$$

makes sense.

Thus if we can compute for each layer $X^i \rightarrow \boxed{W^i} \rightarrow X^{i+1}$

$$\frac{\partial \operatorname{vec}(X^{i+1})}{\partial \operatorname{vec}(W^i)^T}$$ (derivatives of output tensor w.r.t. parameters)

and

$$\frac{\partial \operatorname{vec}(X^{i+1})}{\partial \operatorname{vec}(X^i)^T}$$ (derivatives of output tensor w.r.t. input)

we can compute

$$\frac{\partial z}{\partial \operatorname{vec}(W^i)^T}$$

by working our way <u>backwards</u> from W^L to W^1. This is called back propagation.

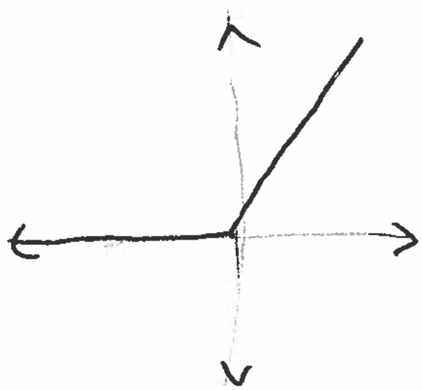
Now we need to Know:

What Kind of layers do we want?

What are their derivatives?

ReLU or Ramp layers.

Consider the function $r(x) = \max(0,x)$.



$$r'(x) = \begin{cases} 1, & x>0 \\ \text{undefined}, & x=0 \\ 0, & x<0 \end{cases}$$

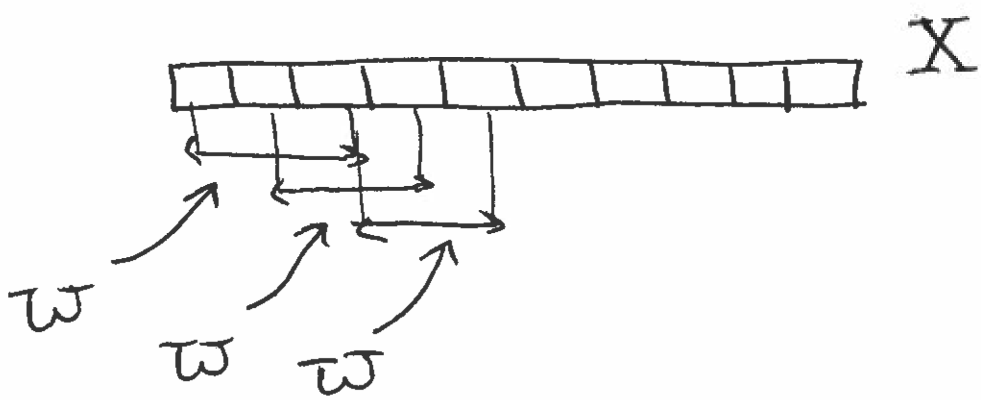
a ramp layer applies $r(x)$ to every entry in X^i, returning an output X^{i+1} of the same shape.

Convolution Layers.

Convolution is an operation ~~with~~ which combines two tensors in a quadratic way.

Example. X is a vector with 10 entries. W is a vector with 3 entries. $X * W$ is a vector with 8 entries defined by

$$(X * W)_i = \underbrace{\sum_{j=1}^{3} W_j X_{i+j-1}}_{\substack{\text{dot product} \\ \text{of } W \text{ with} \\ \underline{\text{part}} \text{ of } X}}$$



The W tensor is called a *convolution kernel*. The ~~convolved~~ *convolution product* $X * W$ is large when X contains *features* detected by W.

Example. Suppose $X_i = f(ih)$ for some differentiable function f and small h, and $W = (1, -2, 1)$.

Them

$$(W * X)_i = X_i - 2X_{i+1} + X_{i+2}$$

$$= f(ih) - 2f((i+1)h) + f((i+2)h).$$

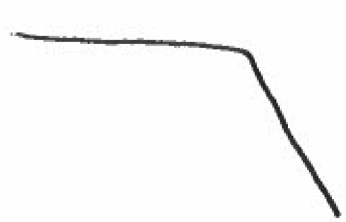
or if $(i+1)h = x_0$,

$$= f(x_0 - h) - 2f(x_0) + f(x_0 + h).$$

To under this term, we expand $f(x_0 \pm h)$ using Taylor's theorem

$$\begin{aligned} &= \cancel{f(x_0)} - \cancel{hf'(x_0)} + \frac{h^2}{2} f''(x_0) - \cancel{\frac{h^3}{6} f'''(x_0)} + \dots \\ &\quad - \cancel{2f(x_0)} \\ &\quad \cancel{f(x_0)} + \cancel{hf'(x_0)} + \frac{h^2}{2} f''(x_0) + \cancel{\frac{h^3}{6} f'''(x_0)} + \dots \\ &= h^2 f''(x_0) + O(h^4) \end{aligned}$$

This is called the Sobel Kernel and it (roughly) detects edges in an image.



Convolution is <u>linear</u> in $\underline{X}$ and <u>linear</u> in W, so the derivatives

$$\frac{\partial(\underline{X} * W)}{\partial \underline{X}}, \quad \frac{\partial(\underline{X} * W)}{\partial W}$$

are easy to compute.

Pooling layer.

The pooling layer replaces disjoint regions of the input tensor with their max ~~of~~ or average. There are no parameters.