Input layer: here a 4x4 matrix of pixel values, X_{ij}

$$X = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & \ddots & & \vdots \\ X_{31} & & \ddots & \vdots \\ X_{41} & \cdots & \cdots & X_{44} \end{bmatrix}$$

$\rightarrow$

Apply filters: 2x2, stride of 1, W_{ij}

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

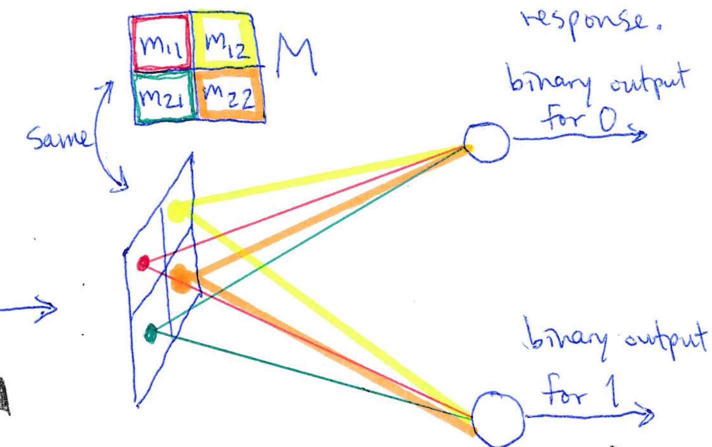
$\rightarrow$

Conv. layer: a 3x3 matrix of input layer convolved w/filter, V_{ij}

$$V = \begin{bmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ V_{31} & V_{32} & V_{33} \end{bmatrix}$$

Max-Pooled layer: applied 2x2 max-pooling to 3x3 conv. layer (stride 1)

Output layer: a 2x1 vector in the case of binary response.



Convolutional error:

- Big change here relative to typical backprop: we look at each value in the original conv layer and use an indicator fctn to identify if that max value is present in that "window", moving L-to-R across and shifting down to start of next row. This gives a square matrix, call it (w^{pool}), of dim. cardinality(max-pool matrix).

Ex: Say orig conv. matrix V is

$$\begin{bmatrix} 3 & 2 & 5 \\ 4 & 1 & 7 \\ 3 & 5 & 0 \end{bmatrix}$$

$\downarrow$

max-pooled matrix M

$$\begin{bmatrix} 4 & 7 \\ 5 & 7 \end{bmatrix}$$

$\downarrow$

reverse max-pool for each submatrix ("window")

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

- For $\sigma'(z^{conv})$, z^{conv} will be a vector, where z_i^{conv} is one of the max-pooled values in the max-pool matrix M. In other words, turn M into a column vector and apply σ' to each element of z^{conv} to get $\sigma'(z^{conv})$.

- FINALLY:

$$\underset{4\times1}{\delta^{conv}} = \Big[\underset{4\times4}{(w^{pool})^T}\,\underset{4\times1}{(\delta^{pool})}\Big] \odot \Big[\underset{4\times1}{\sigma'(z^{conv})}\Big]$$

Max-Pool Error:

- Same as for a FF-NNet, but we vectorize the weights connecting each m_{ij} and the output nodes. Thus:
- # rows in W^L = 4 = # elements in M.
- # cols in W^L = 2 = # nodes in output layer.
- FINALLY:

$$\underset{4\times1}{\delta^{pool}} = \Big[\underset{4\times2}{(W^L)^T}\,\underset{2\times1}{(\delta^L)}\Big] \odot \Big[\underset{4\times1}{\sigma'(z^{pool})}\Big]$$

Output error:

- Same as for a FF-NNet

$$\underset{2\times1}{\delta^L} = \Big[\underset{2\times1}{\nabla_{a^L} C}\Big] \odot \Big[\underset{2\times1}{\sigma'(z^L)}\Big]$$

- **NOTE:** if we turn the 2x2 filter into a 4x1 vector we see that $[\dim(\text{filter}) = 4\times1] == [\dim(\delta^{conv}) = 4\times1]$. Dimensions align!