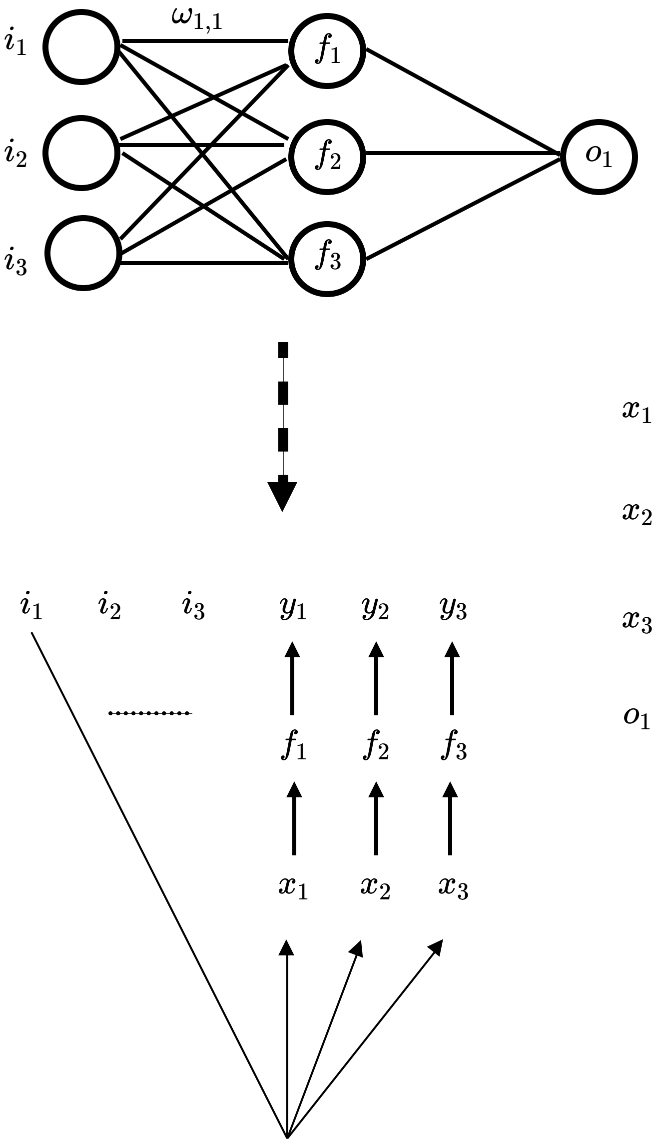
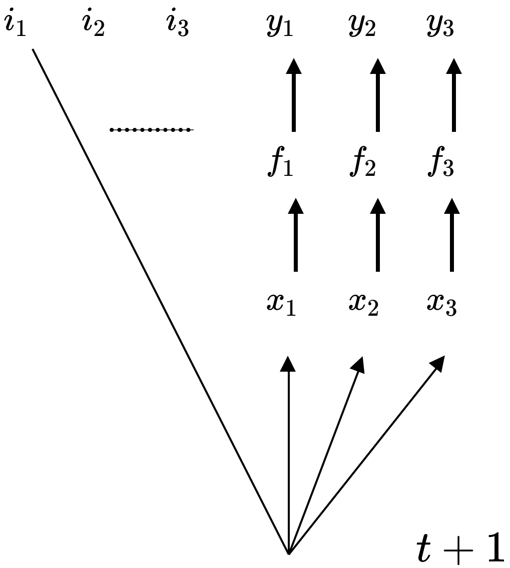


$W_{m,n}$ Connectivity / weight matrix, transforms neuron outputs @ time t to neuron inputs @ time t+1 (down step)

$$
\begin{array}{c|cccccc|}
 & i_1 & i_2 & i_3 & y_1 & y_2 & y_3 \\
\hline
x_1 & \omega_{1,1} & & & 0 & 0 & 0 \\
x_2 & & & & 0 & 0 & 0 \\
x_3 & & & & 0 & 0 & 0 \\
o_1 & 0 & 0 & 0 & & & \omega_{4,6} \\
\hline
\end{array}
\otimes
\overset{t}{\begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}}
=
\overset{t+1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ o_1 \end{bmatrix}}
$$

f_i Non linear (squashing) functions transform neuron inputs @ time t-1 to neuron outputs @ time t (up step)

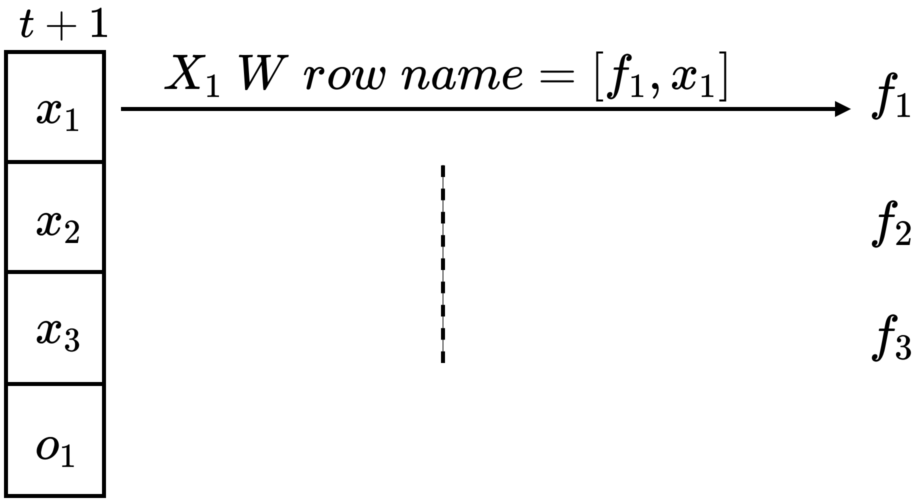


What we have shown so far gives the machinery for the *down*-step, which is basically matrix multiplication.

$W_{m,n}$ Connectivity / weight matrix, transforms neuron outputs @ time t to neuron inputs @ time t+1 (down step)

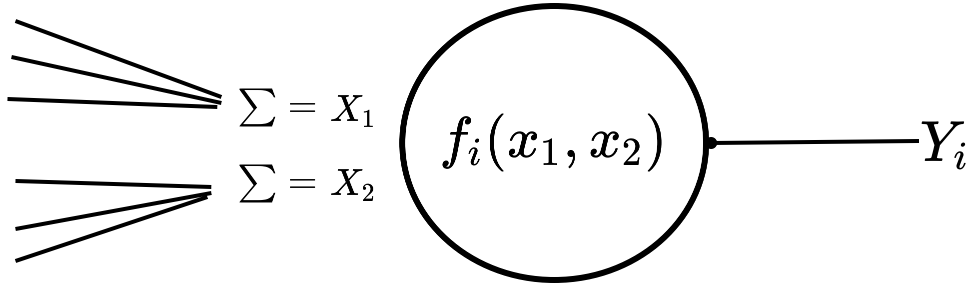
But we need another piece for the *up*-step. Encode which input goes to which neuron. For illustration, here neurons are identified by their activation function name.

Row names are structures encoding neuron name and corresponding activation function argument name.

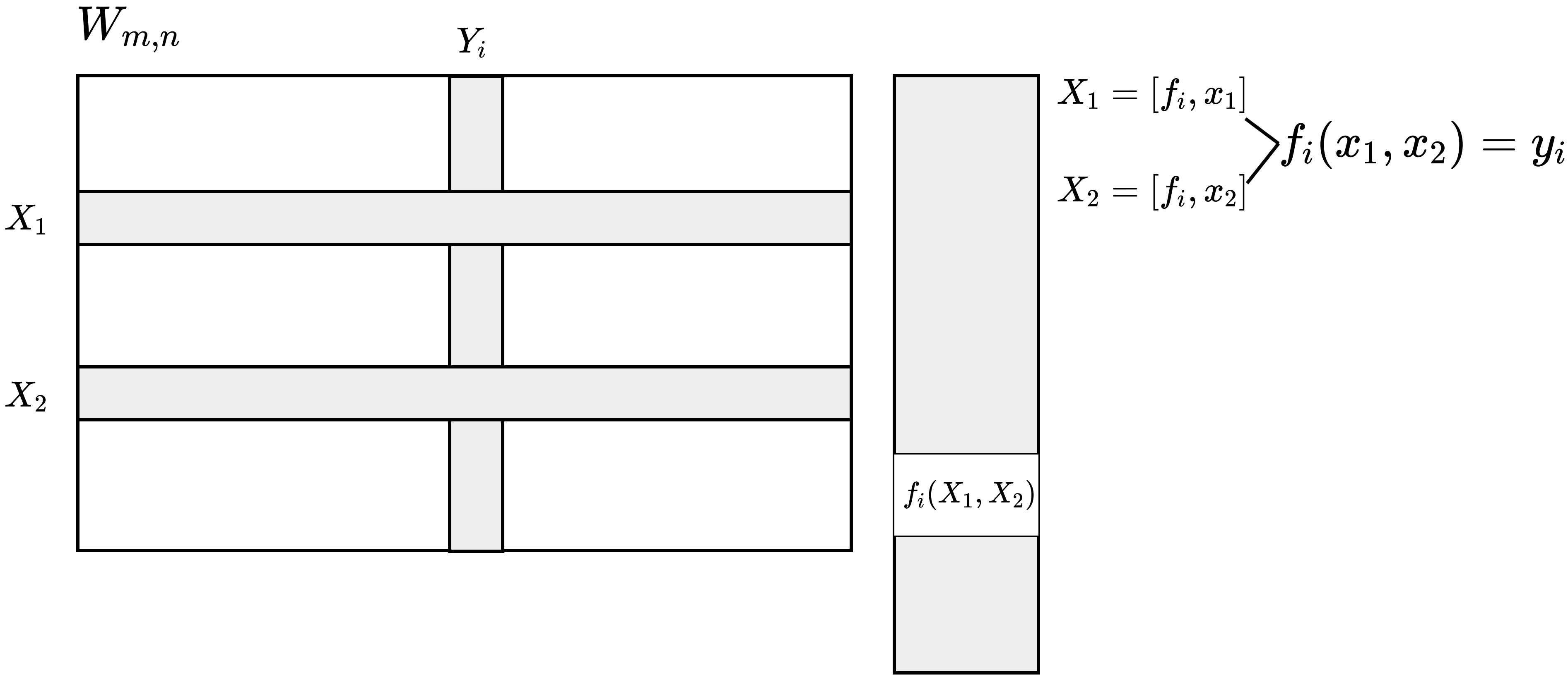


f_i Non linear (squashing) functions transform neuron inputs @ time t-1 to neuron outputs @ time t (up step)

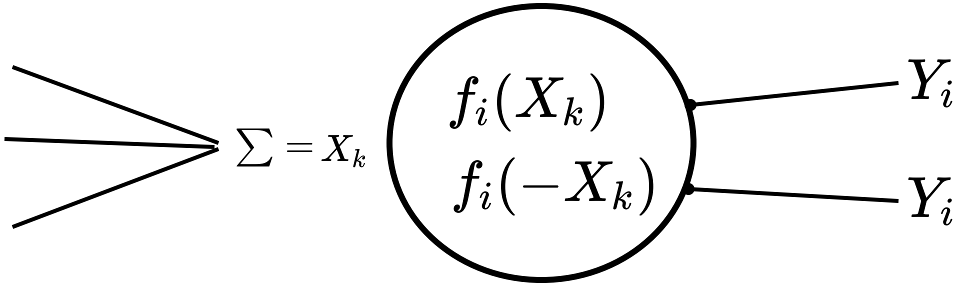
Variadic Neurons - input

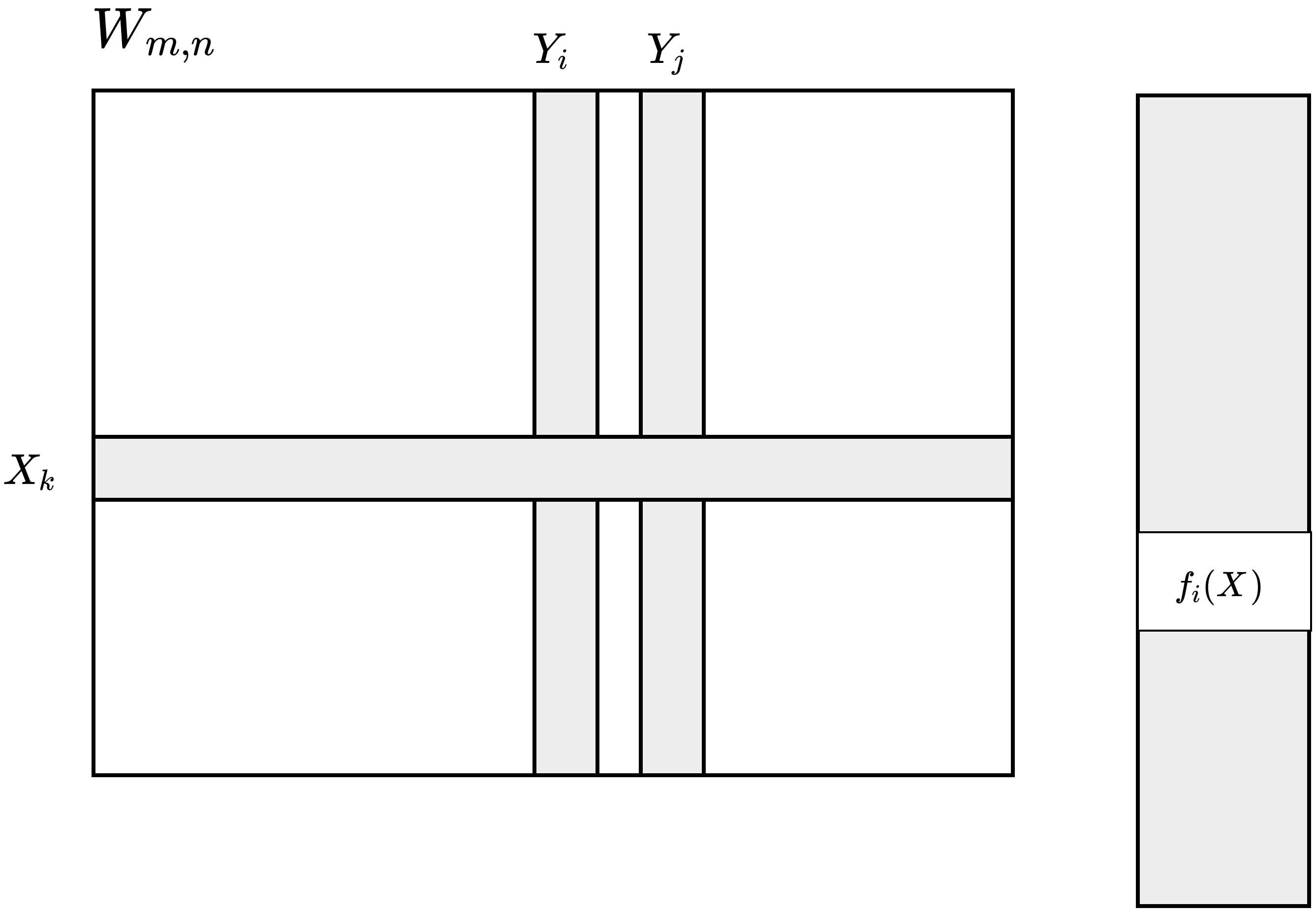


For example, *f* here could be element wise multipliation giving an attenuating memory gate (as in LSTM)



Variadic Neurons - output





Accumulator