Multi-Layer Neural Networks
[RN2] Sec 20.5
[RN3] Sec 20.5

CS 486/686

University of Waterloo

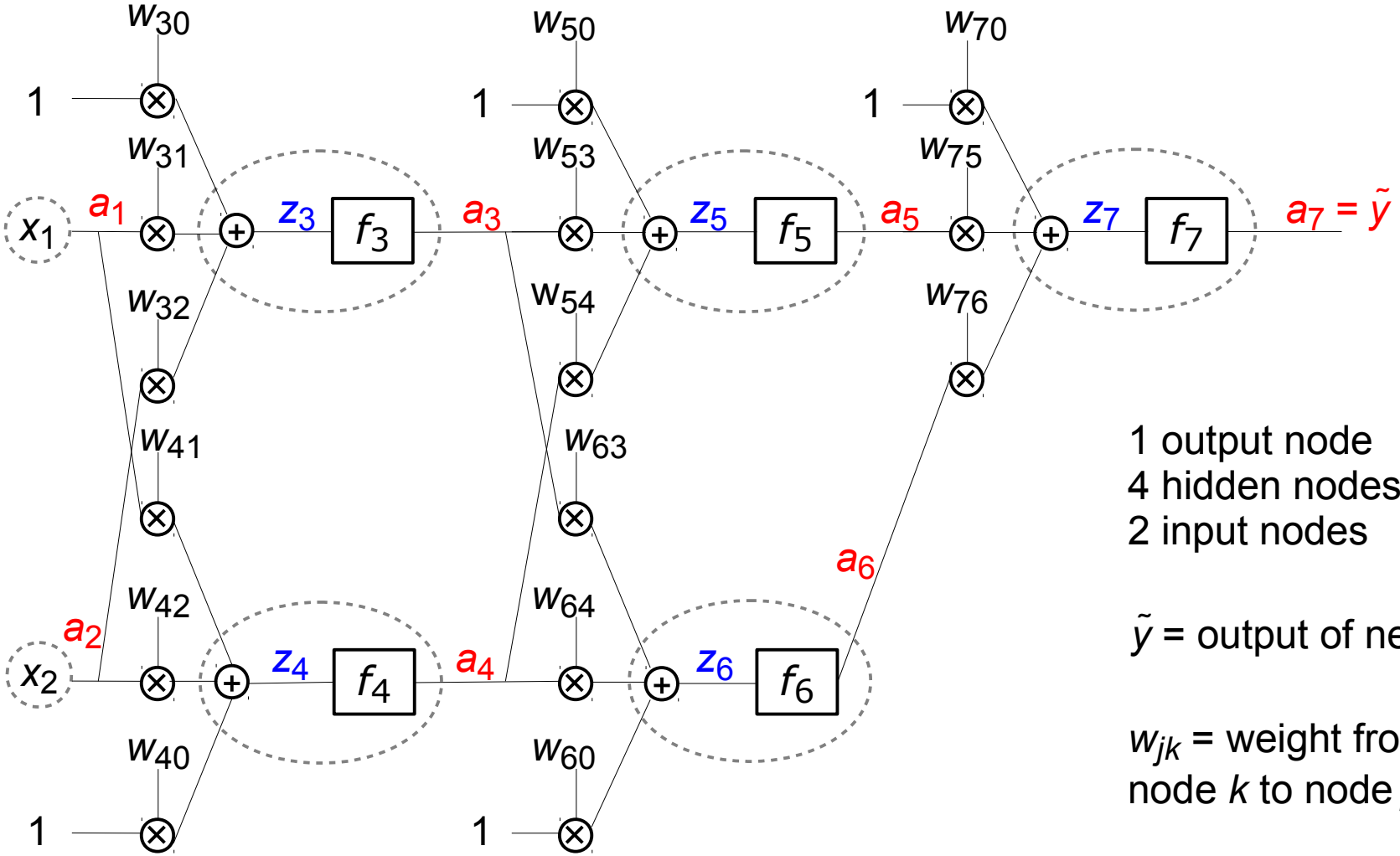
Lecture 20: July 9, 2015

Multi-Layer Neural Networks

- Perceptron can only represent linear separators

- Need multiple layers to represent more complicated separators

Example: Two Hidden Layers



1 output node
4 hidden nodes
2 input nodes

$\tilde{y}$ = output of network

w_{jk} = weight from node k to node j

Learning Multi-Layer Network

- Minimize squared error:

$$E(w, x, y) = \frac{1}{2}(y - \tilde{y})^2$$

$$\hat{w} = \arg\min_{w} E(w, x, y)$$

- Solution: gradient descent
- Just like what we did with sigmoid perceptron!

Learning Multi-Layer Network

- Minimize squared error:

$$E(w, x, y) = \frac{1}{2}(y - \tilde{y})^2$$

$$\hat{w} = \arg\min_{w} E(w, x, y)$$

- Solution: gradient descent
- Just like what we did with sigmoid perceptron!
- Problem: gradient much harder to compute

Learning Multi-Layer Network

- Minimize squared error:

$$E(w, x, y) = \frac{1}{2}(y - \tilde{y})^2$$

$$\hat{w} = \arg\min_{w} E(w, x, y)$$

- Solution: gradient descent
- Just like what we did with sigmoid perceptron!
- Problem: gradient much harder to compute
  - Solution: compute gradient with backpropagation

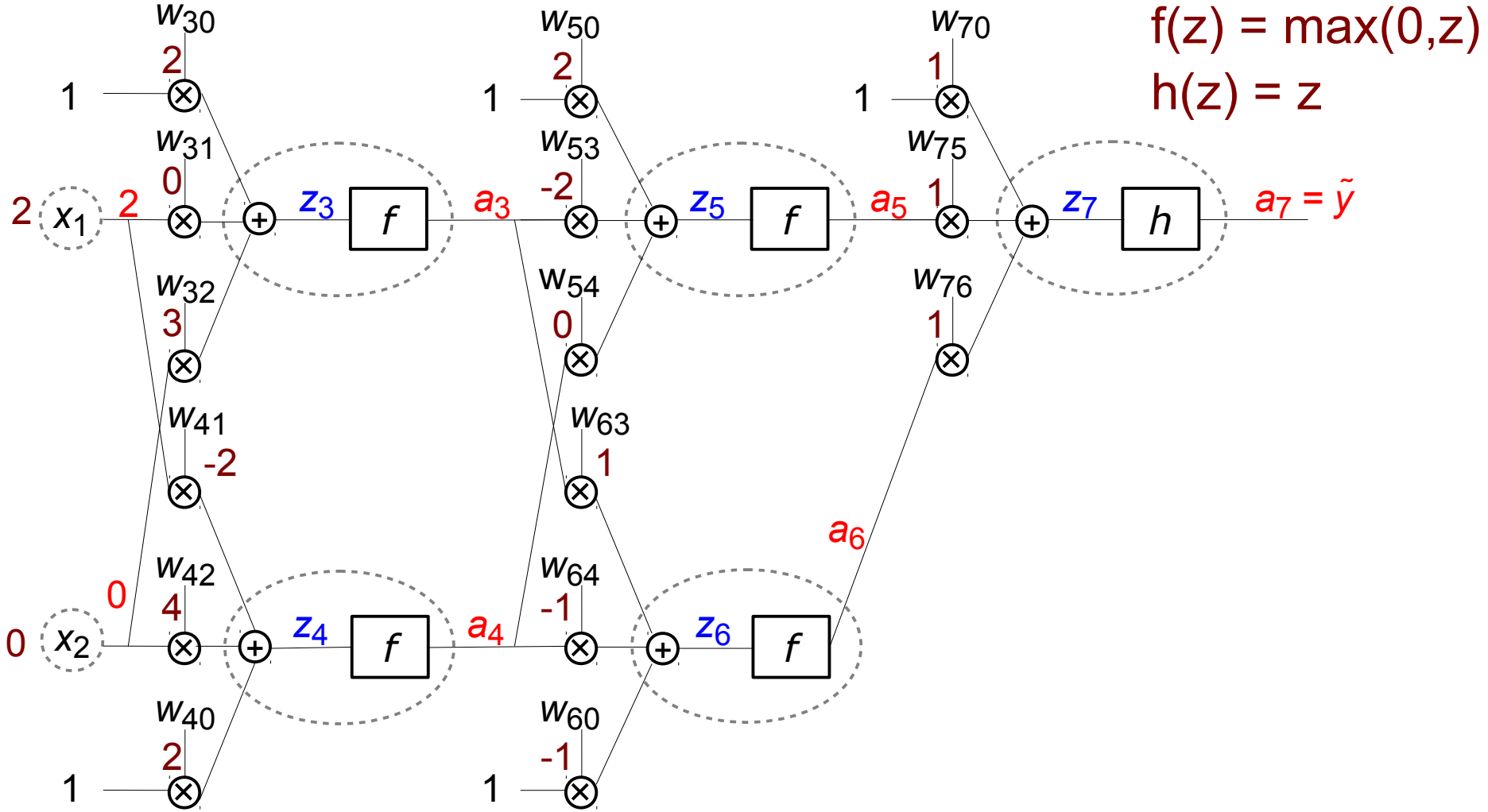
Learning Multi-Layer Network by Gradient Descent

- Initialize weights *w*
- For each training example (***x***,*y*) do
  - Compute gradient $\nabla E(w)$ by backpropagation
  - Update weights $w \leftarrow w - \alpha \nabla E(w)$
- Repeat until stopping criteria satisfied

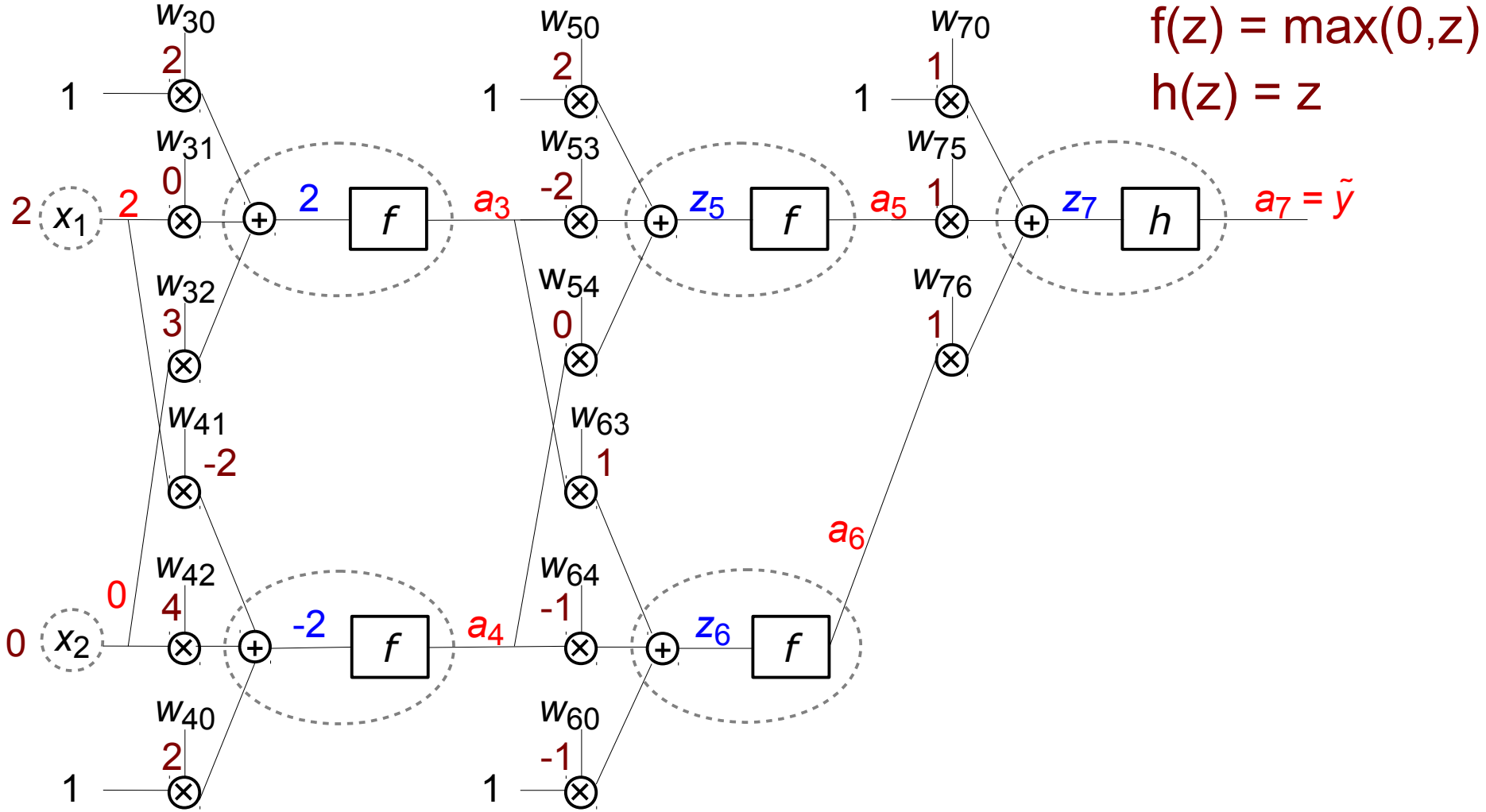
Backpropagation

- Given training example (*x*, *y*)
- Forward phase
  - Starting from input nodes, compute all z_k's and a_k's by forward propagation
- Backward phase
  - Starting from output nodes, compute $D_k = \frac{\partial E}{\partial z_k}$:
    - $D_k = f'(z_k)(y_k - a_k)$ if *k* is an output node
    - $D_k = f'(z_k)\sum_j w_{jk} D_j$ if *k* is a hidden node
  - Compute all weight derivatives $\frac{\partial E}{\partial w_{jk}} = D_j a_k$
- Return $\nabla E(w)$

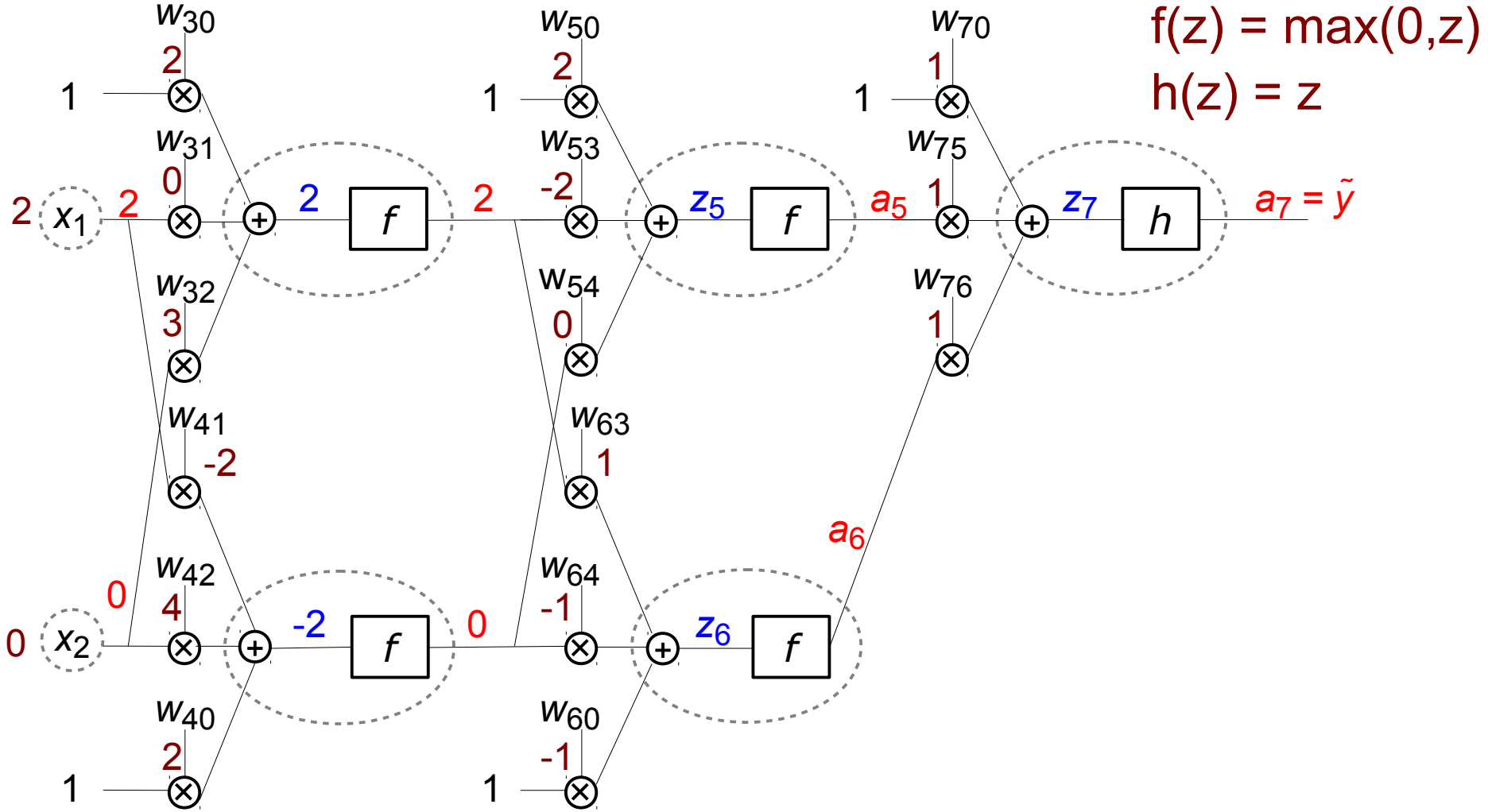
Example: Forward Phase



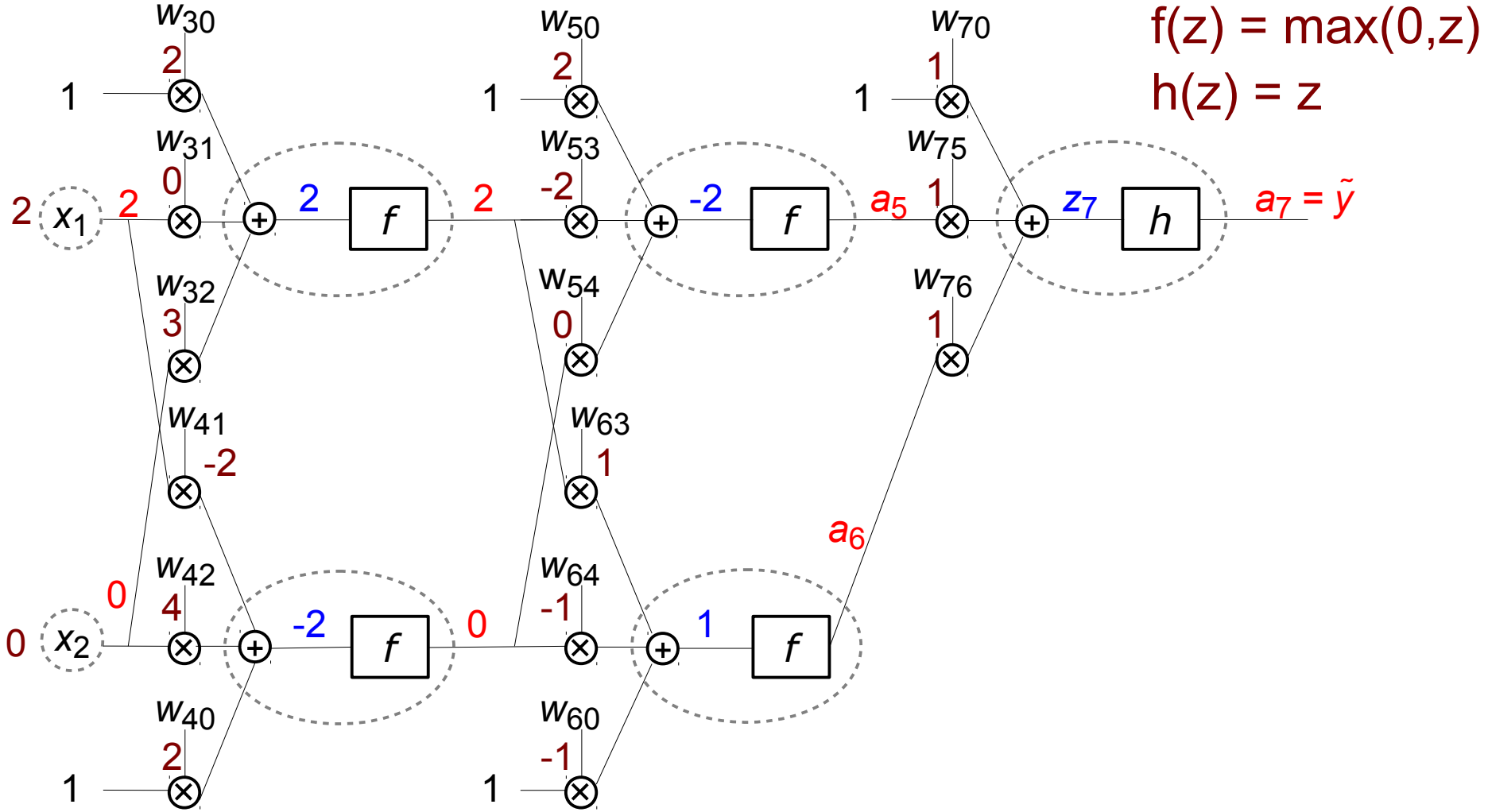
Example: Forward Phase



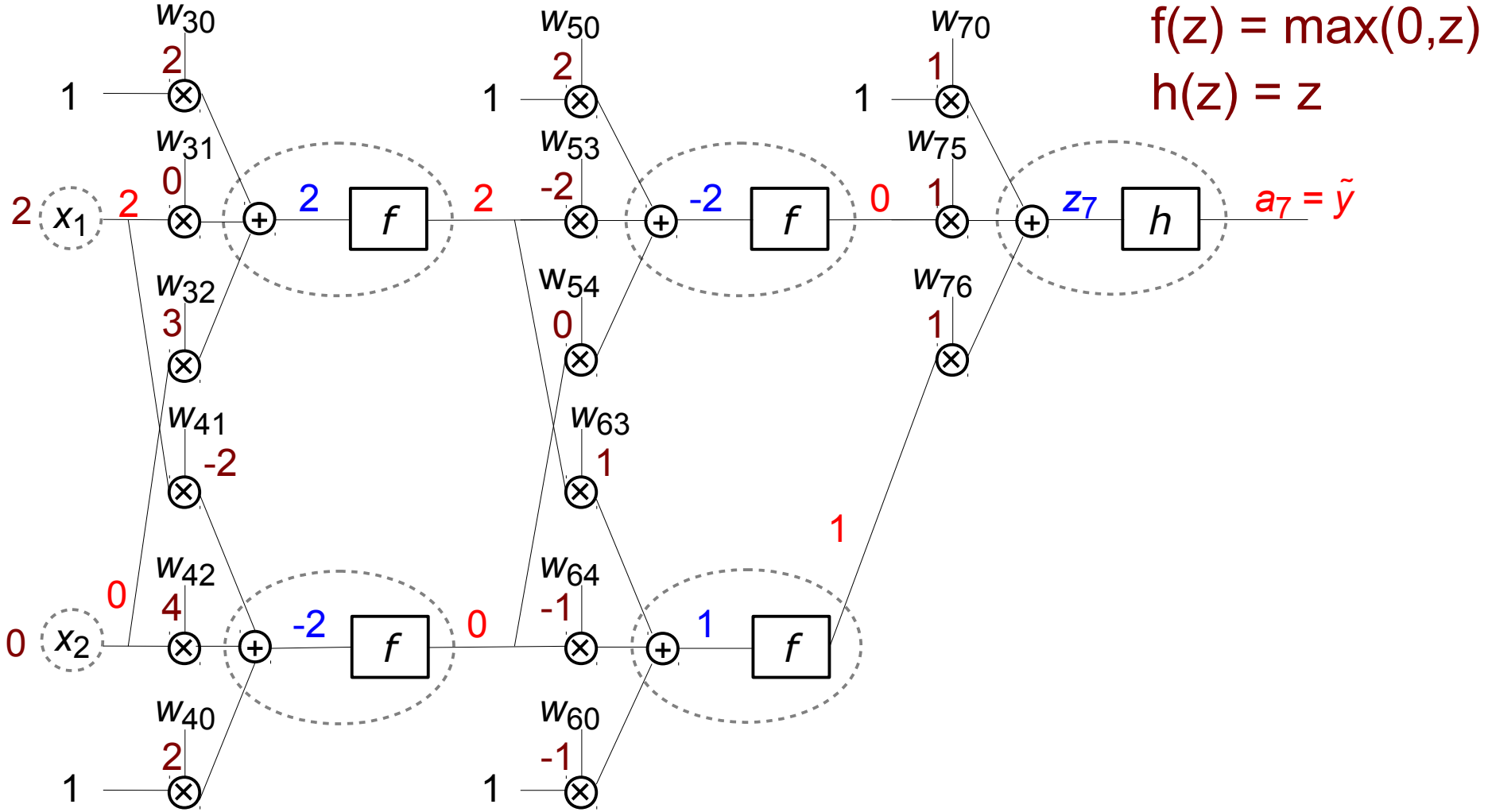
Example: Forward Phase



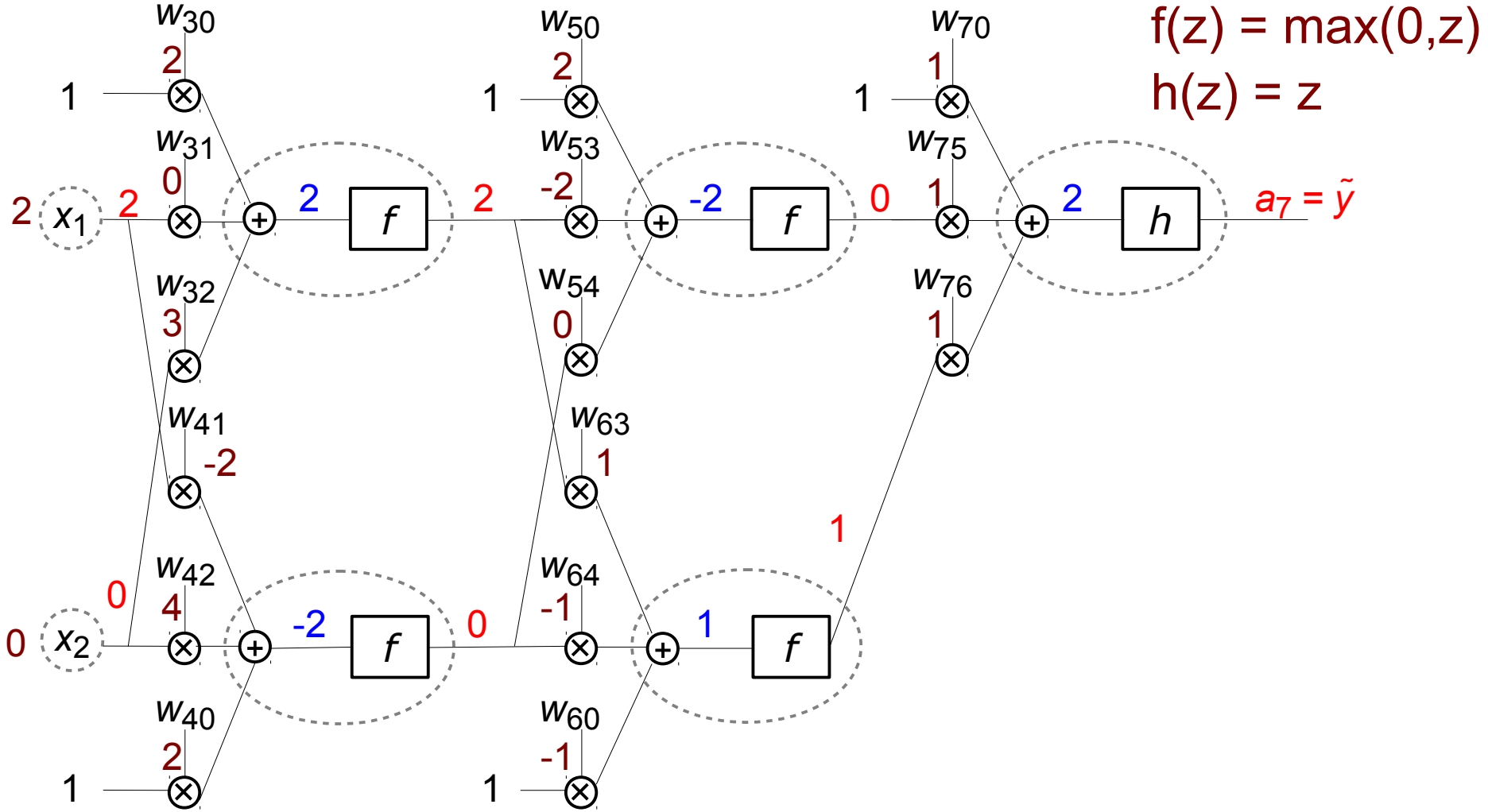
Example: Forward Phase



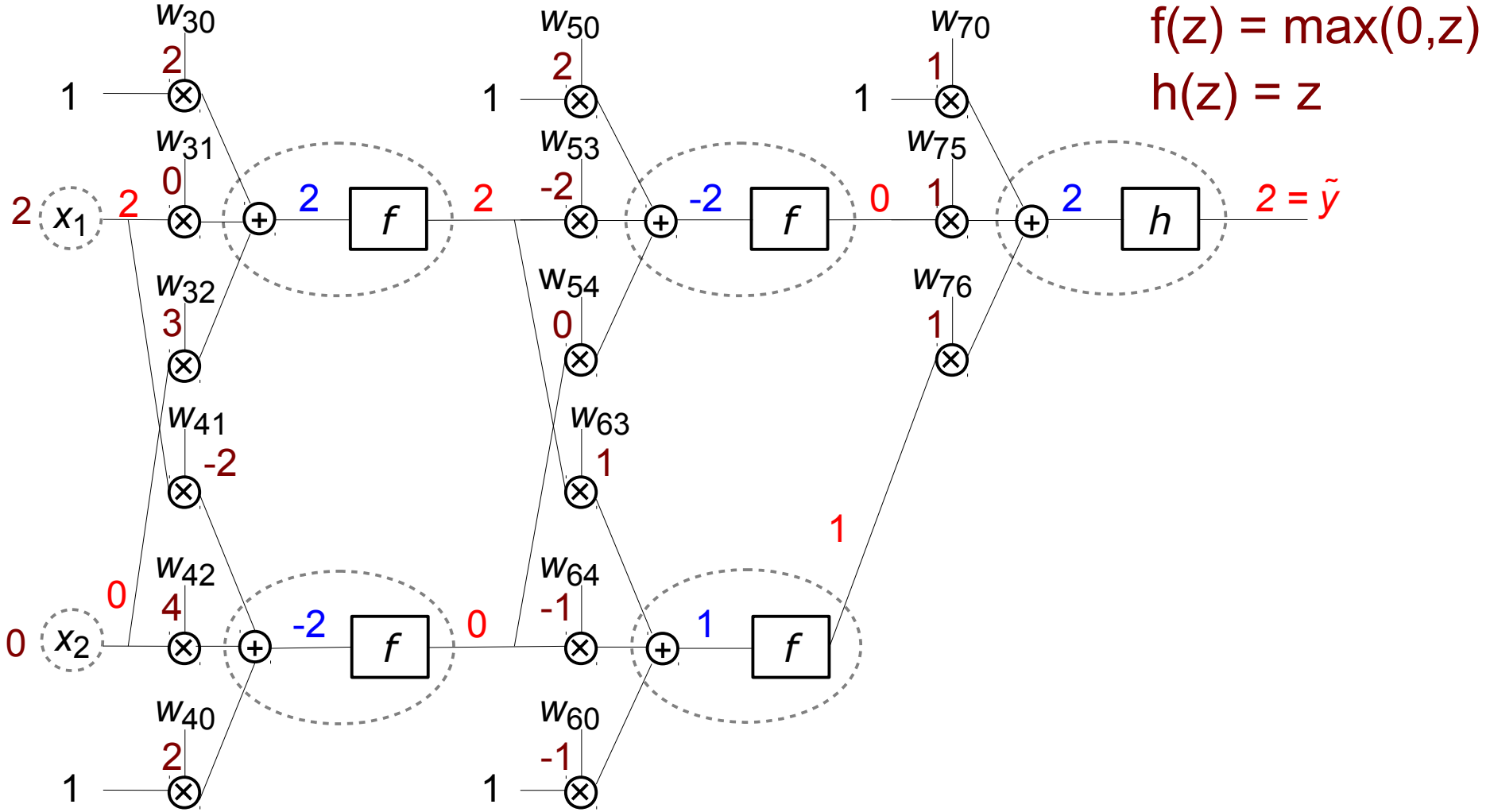
Example: Forward Phase



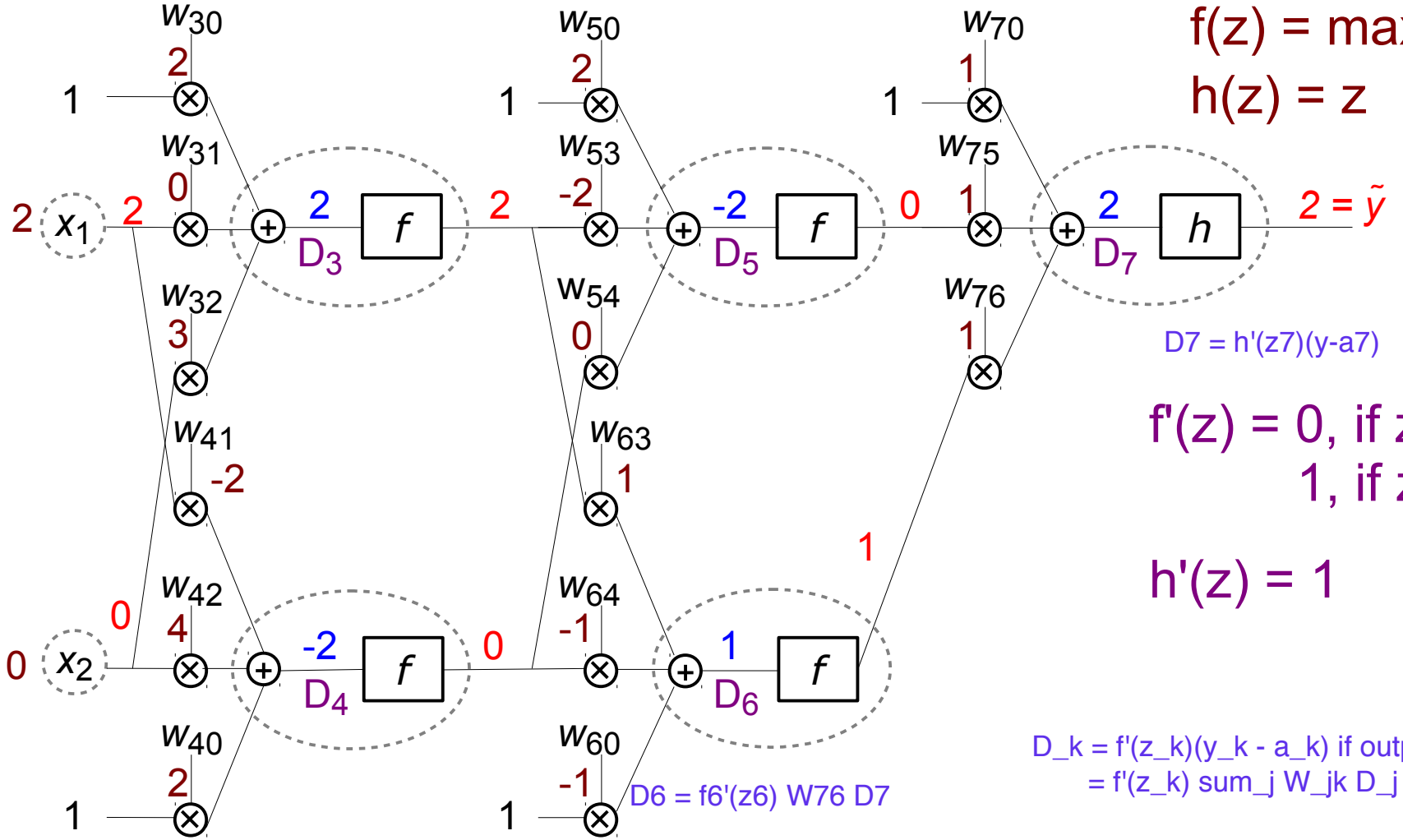
Example: Forward Phase



Example: Forward Phase



Example: Backward Phase



f(z) = max(0,z)

h(z) = z

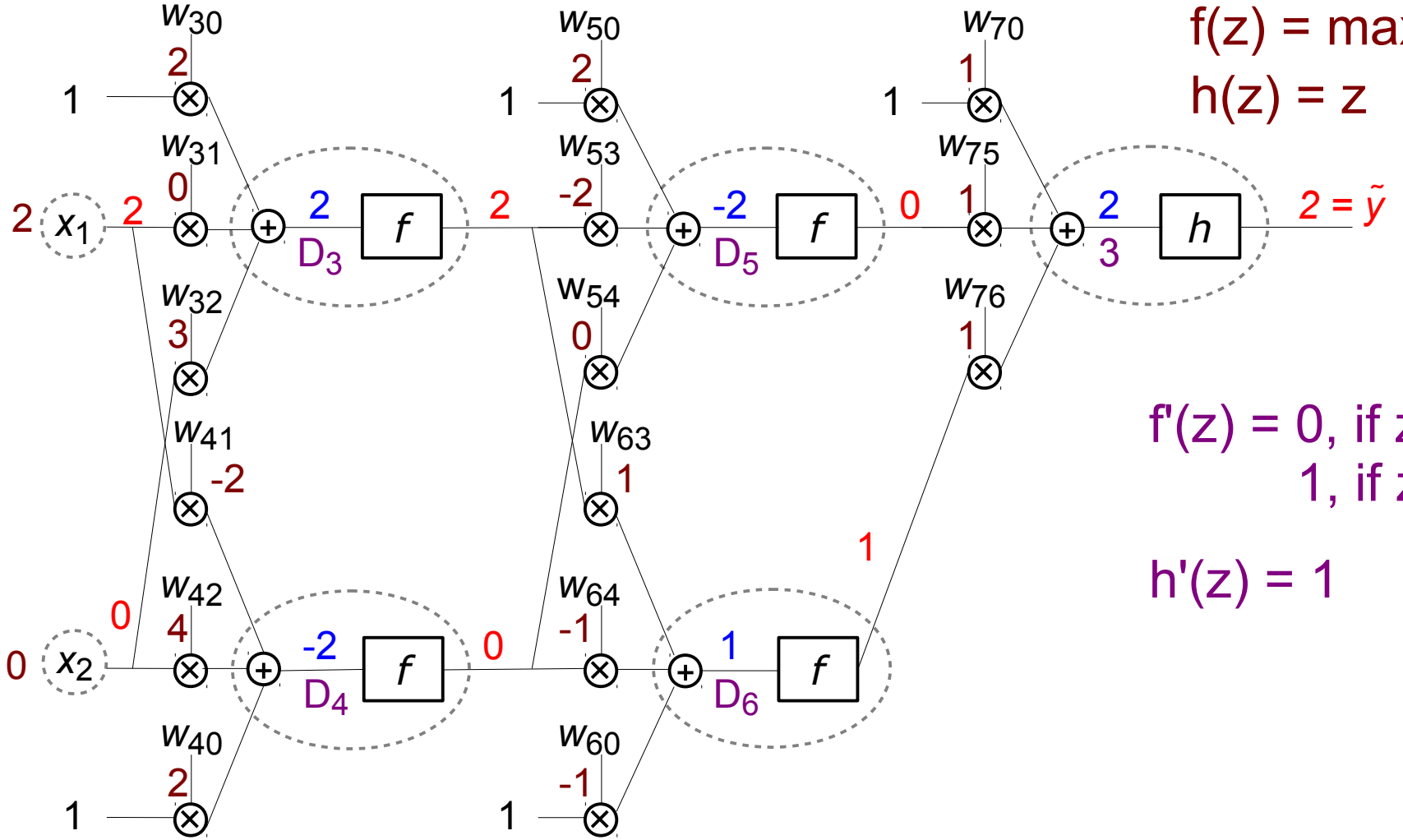
D7 = h'(z7)(y-a7)

f'(z) = 0, if z < 0
1, if z > 0

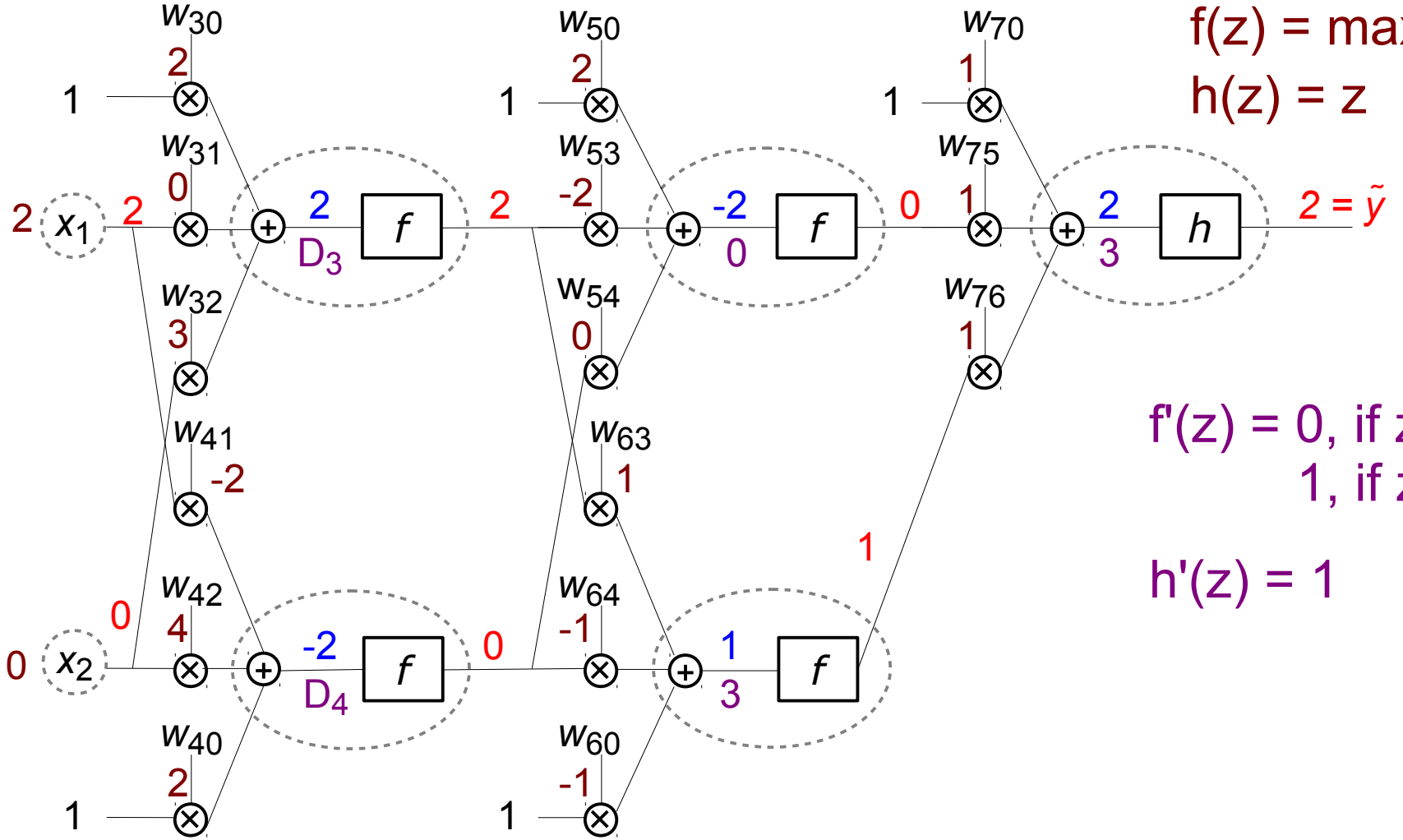
h'(z) = 1

D_k = f'(z_k)(y_k - a_k) if output
= f'(z_k) sum_j W_jk D_j

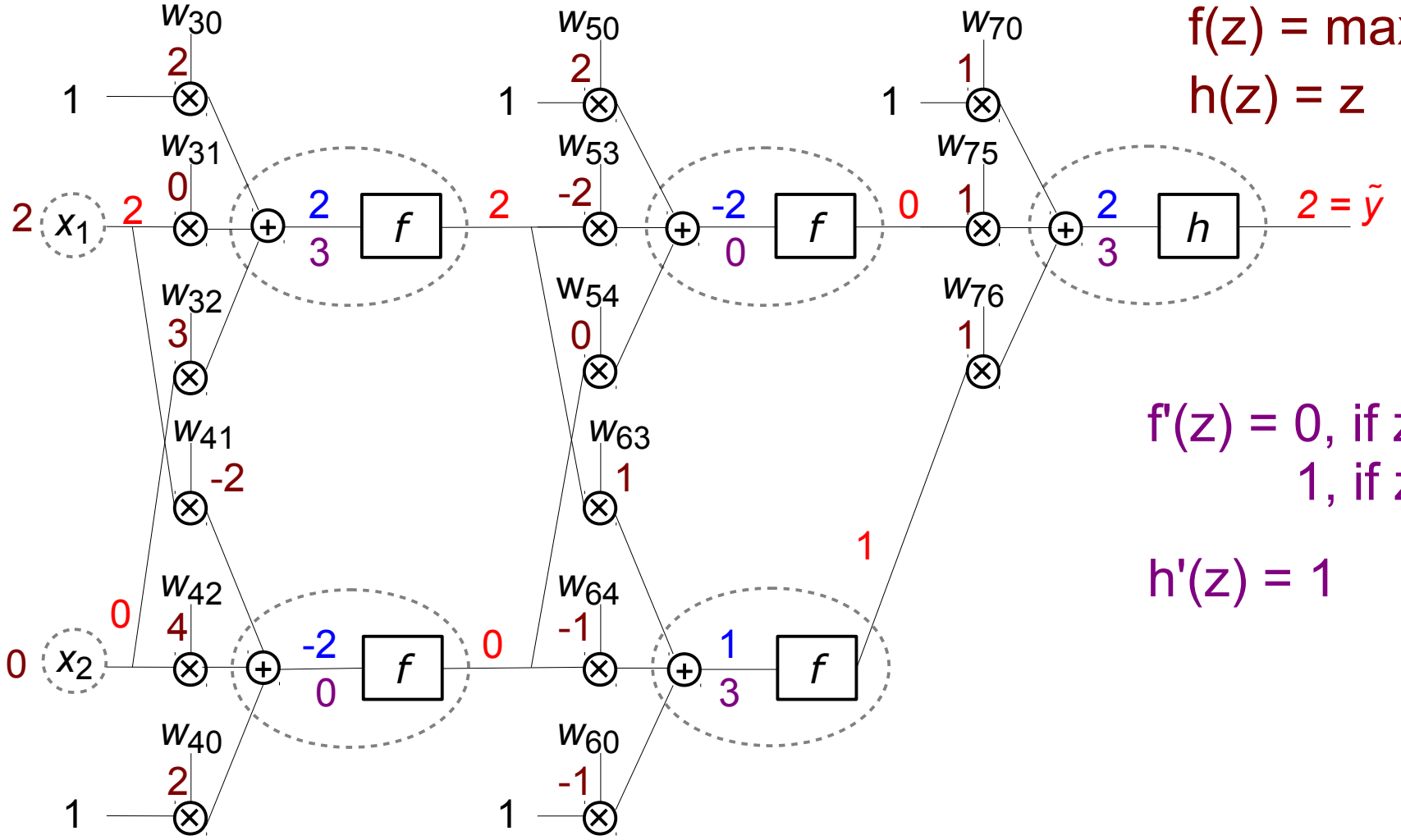
Example: Backward Phase



Example: Backward Phase



Example: Backward Phase



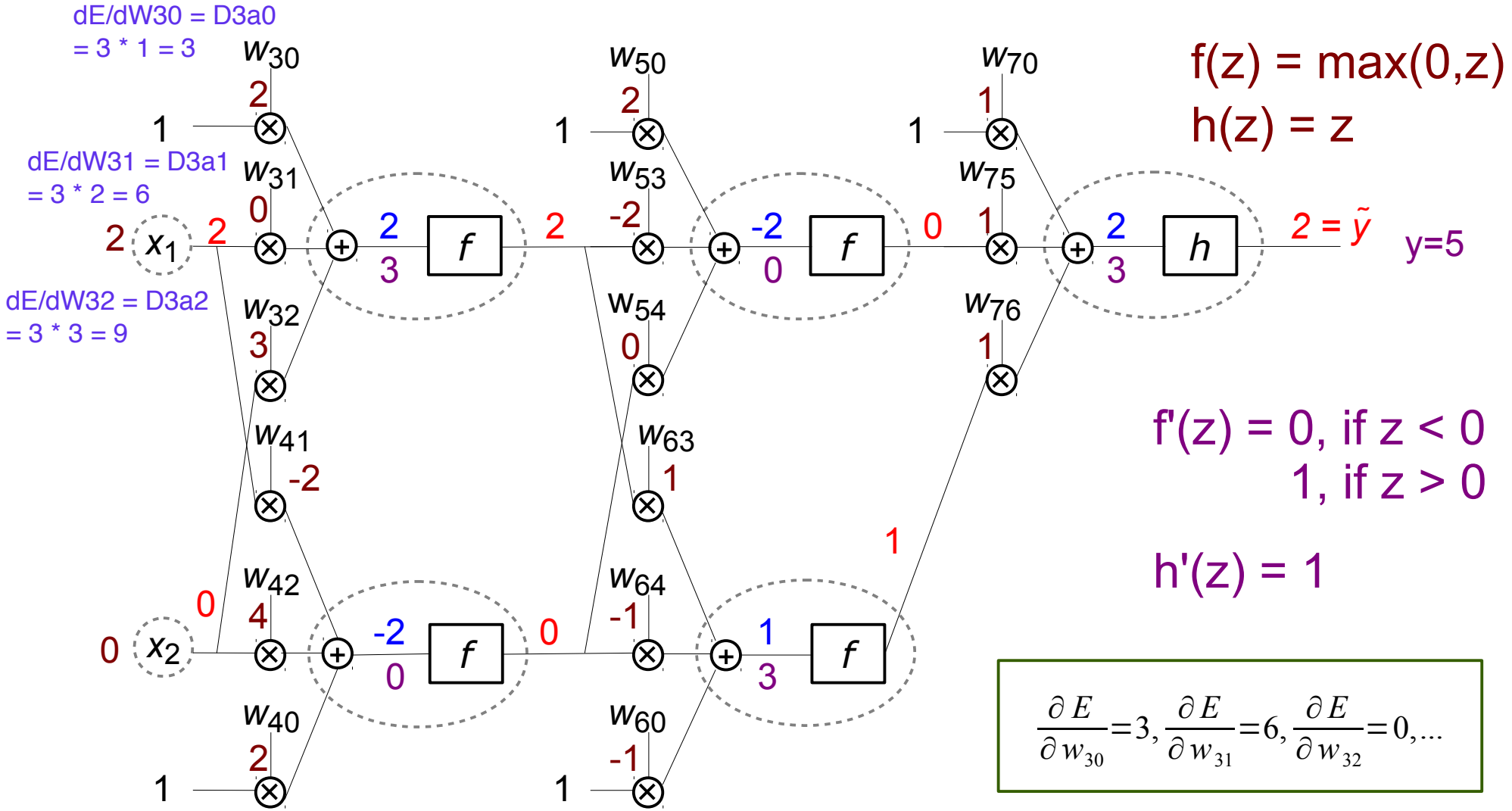
$f(z) = \max(0,z)$

$h(z) = z$

$f'(z) = 0$, if $z < 0$
1, if $z > 0$

$h'(z) = 1$

Example: Backward Phase



f(z) = max(0,z)

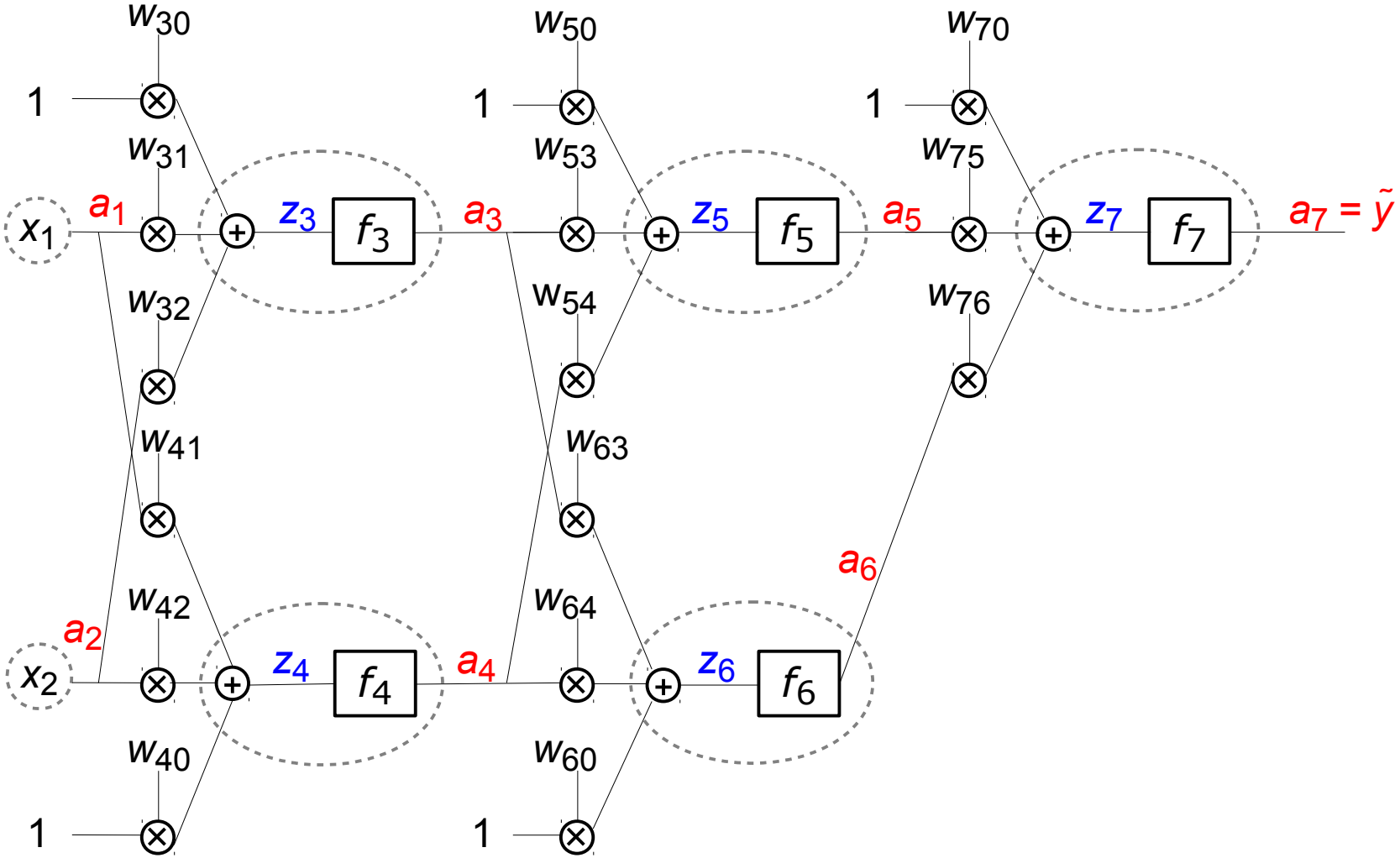
h(z) = z

f'(z) = 0, if z < 0
1, if z > 0

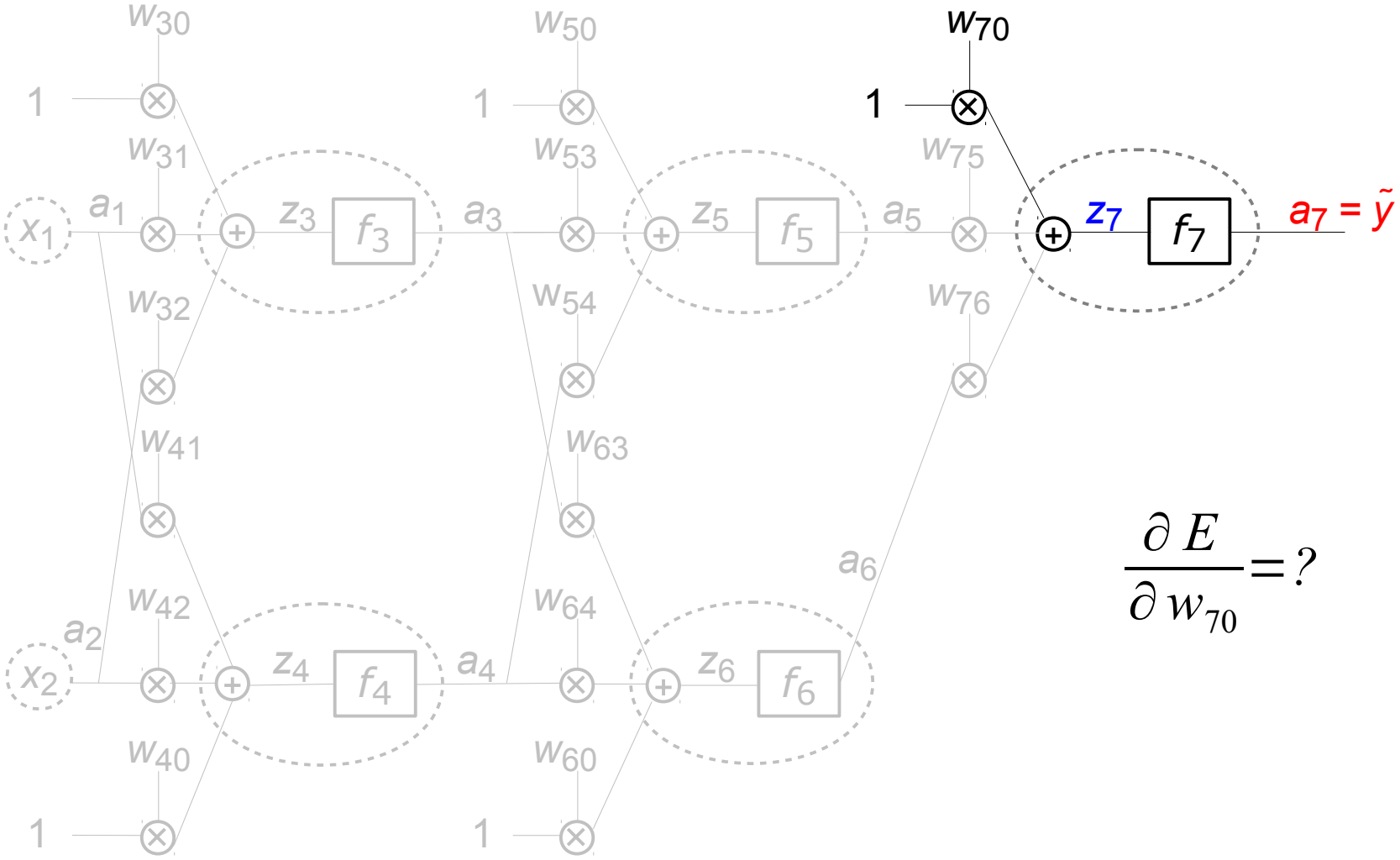
h'(z) = 1

$$\frac{\partial E}{\partial w_{30}}=3,\ \frac{\partial E}{\partial w_{31}}=6,\ \frac{\partial E}{\partial w_{32}}=0,\ldots$$

Backpropagation Derivation

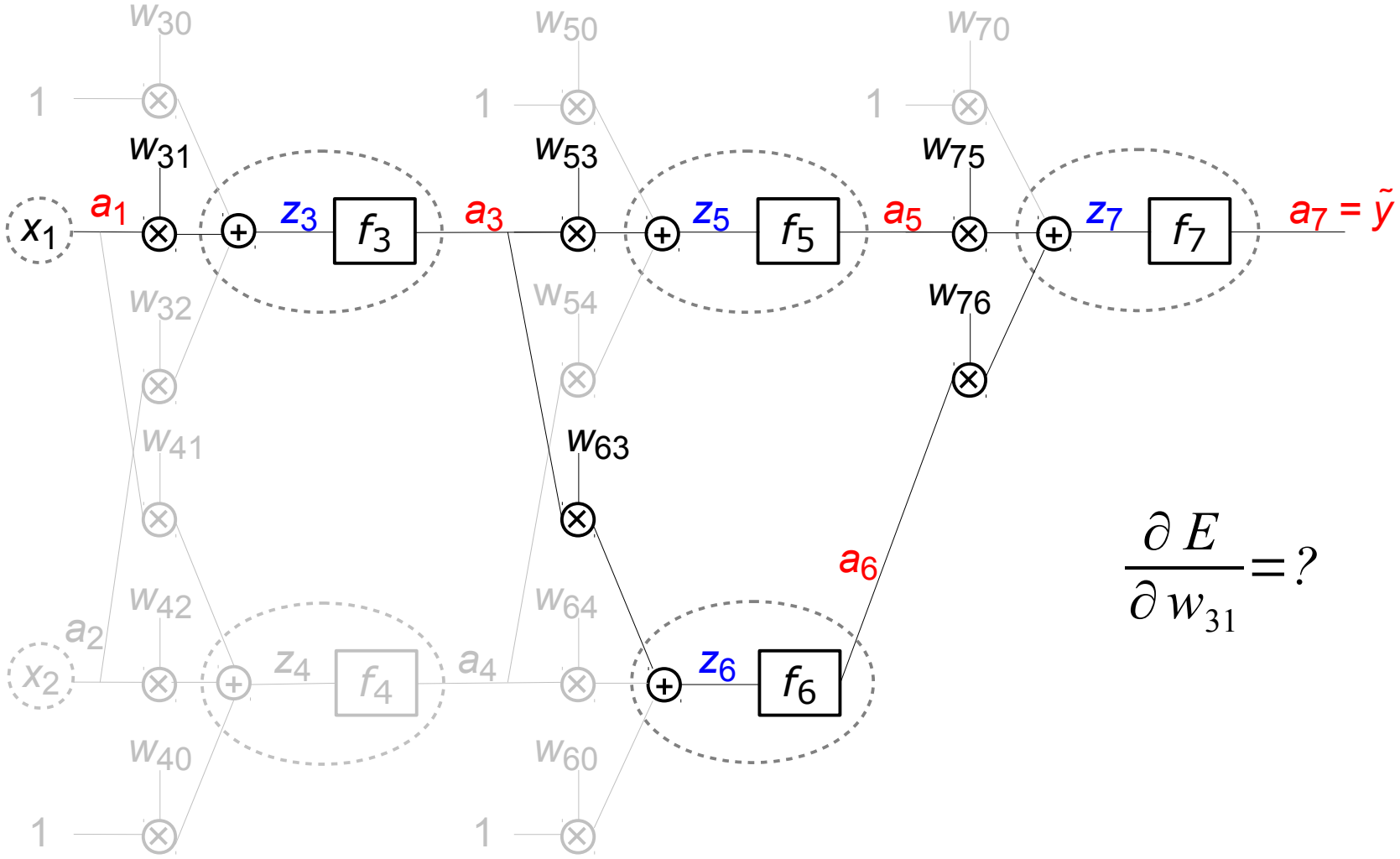


Backpropagation Derivation



$$\frac{\partial E}{\partial w_{70}} = ?$$

Backpropagation Derivation



$$\frac{\partial E}{\partial w_{31}} = ?$$