Deep Learning

CS256 - Topics in Artificial Intelligence

February 12, 2018

Overview

Gradient Descent

- write down $\frac{\partial}{\partial x_i} f(x_1, x_2, \dots)$ for all x_i
- run the algorithm:

1. Start at location x_0
2. Set
   - 2.1 iteration counter $i = 0$
   - 2.2 exit condition $exit = False$
   - 2.3 Error threshold θ
   - 2.4 Learning rate η
   - 2.5 Maximum number of iterations $Max_{Iterations}$
3. while $exit$ is $False$
   - 3.1 Compute $\frac{\partial}{\partial x_i} f(\mathbf{x_i})$
   - 3.2 Set $\mathbf{x}_{i+1} = x_0 - \eta \cdot \nabla f(\mathbf{x_i})$
   - 3.3 If $\|\nabla f(\mathbf{x_i})\| < \theta$, set $exit = True$
   - 3.4 If $i > Max_{Iterations}$, set $exit = True$

Stopping criterion

1. When the maximum number of iterations has been reached
2. When we think we are in a local minimum
   - Local minimum can be detected by one of 2 methods
     - The difference from the previous step to the current step is close to 0
     $$\left|f\left(\mathbf{x}^{t-1}\right) - f\left(\mathbf{x}^{t}\right)\right| < \theta_1$$
     - The gradient is close to 0
     remember Taylor approximation $f(x+\epsilon) \approx f(x) + f'(x)\epsilon$
     $$\begin{aligned}\left|f\left(\mathbf{x}^{t-1}\right) - f\left(\mathbf{x}^{t}\right)\right| &= \left|f\left(\mathbf{x}^{t}\right) - f\left(\mathbf{x}^{t} - \eta\nabla f(\mathbf{x})\right)\right| \\ &\approx \left|f\left(\mathbf{x}^{t}\right) - f\left(\mathbf{x}^{t}\right) - \nabla f(\mathbf{x})\eta\nabla f(\mathbf{x})\right| \\ &= \eta\nabla^2 f(\mathbf{x})\end{aligned}$$
     - i.e. instead of comparing the function value at the previous step to the current value, we can consider the gradient directly
     $$\|\nabla f\left(\mathbf{x}\right)\| < \theta_2$$

Notation remark

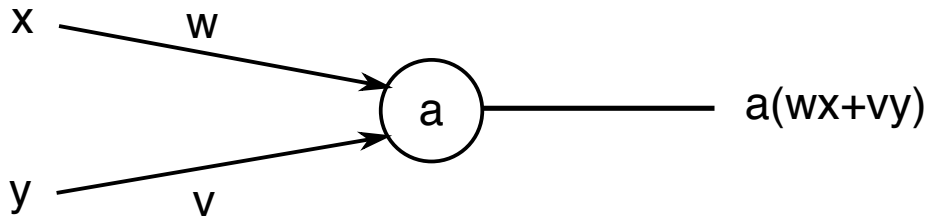
Difference between $|\cdot|$ and $\|\cdot\|$:

- I use $|\cdot|$ exclusively for the absolute value $|-3| = 3$
- In several dimensions, we can consider the following norms:

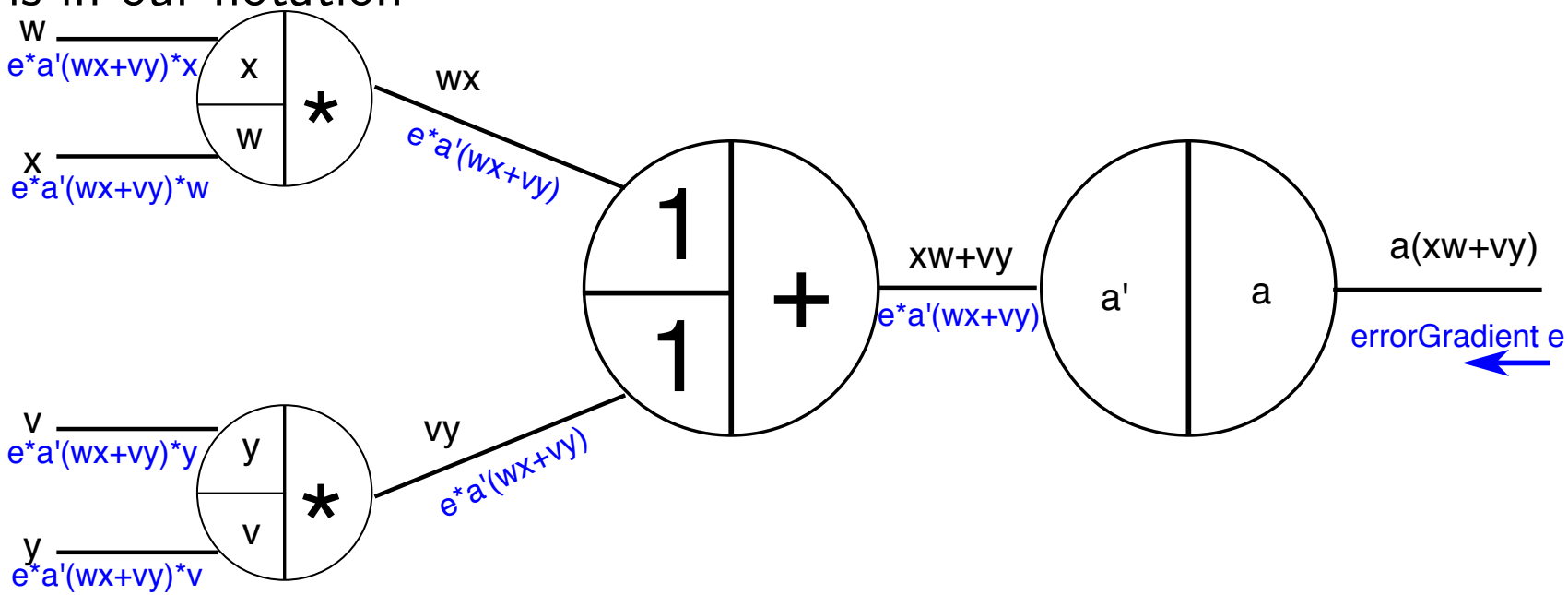
$$\begin{aligned}
\|(x_1, x_2, \dots)\|_k &= \sqrt[k]{x_1^k + x_2^k + \dots} \\
\|(x_1, x_2, \dots)\|_1 &= |x_1| + |x_2| + \dots \quad \text{(L1 Norm)} \\
\|(x_1, x_2, \dots)\|_2 &= \sqrt{x_1^2 + x_2^2 + \dots} \quad \text{(L2 Norm)} \\
\|(x_1, x_2, \dots)\|_\infty &= \max_i x_i \quad \text{(Infinity Norm)}
\end{aligned}$$

- Using $\|\cdot\|$ without and index exclusively means $\|\cdot\|_2$
  - Some people use $|\cdot|$ for this, because this is the normal vector norm (length of a vector)

Back-Propagation

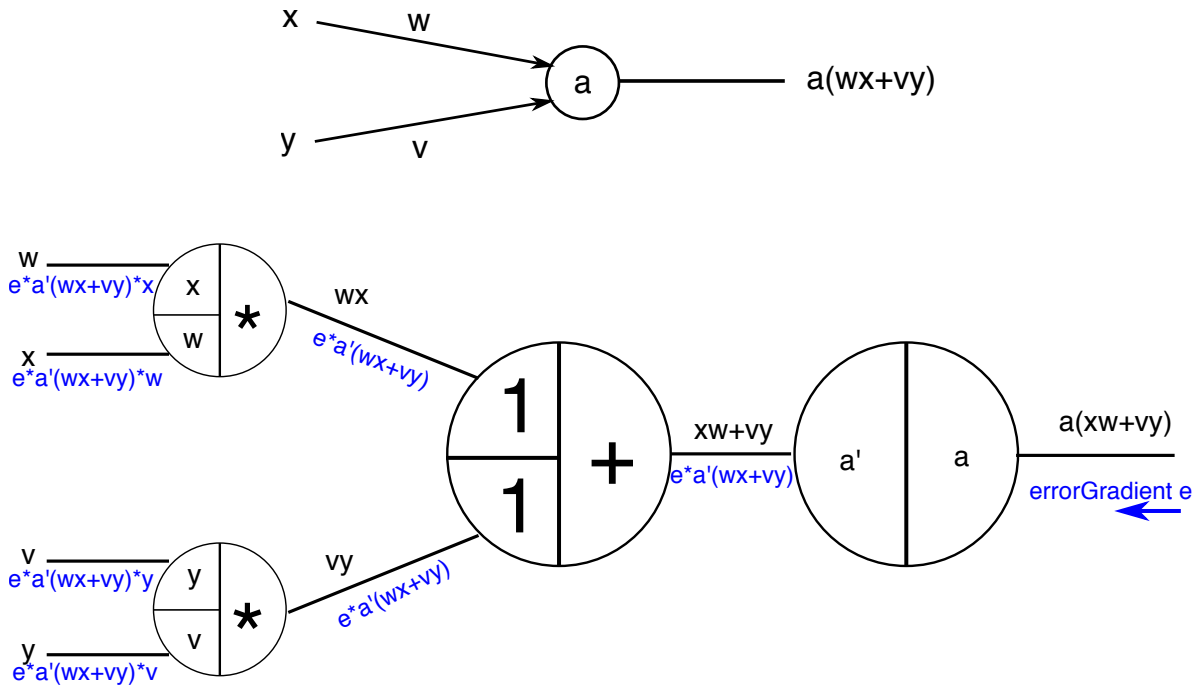


is in our notation



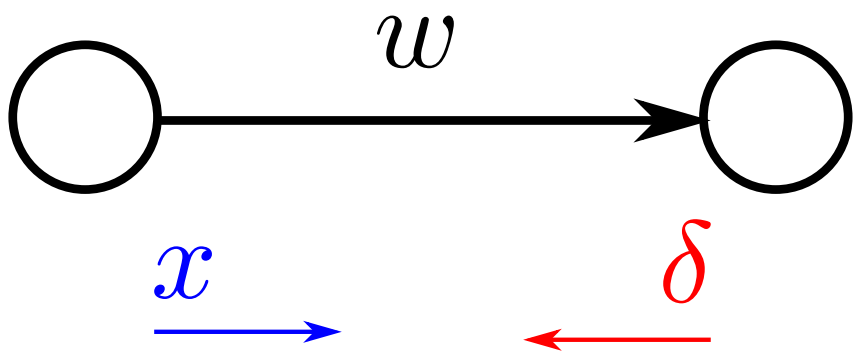
Consider weight w and input x.

Back-Propagation



Consider weight w and input x. The gradient of the error at w is the error gradient times x

Back-Propagation

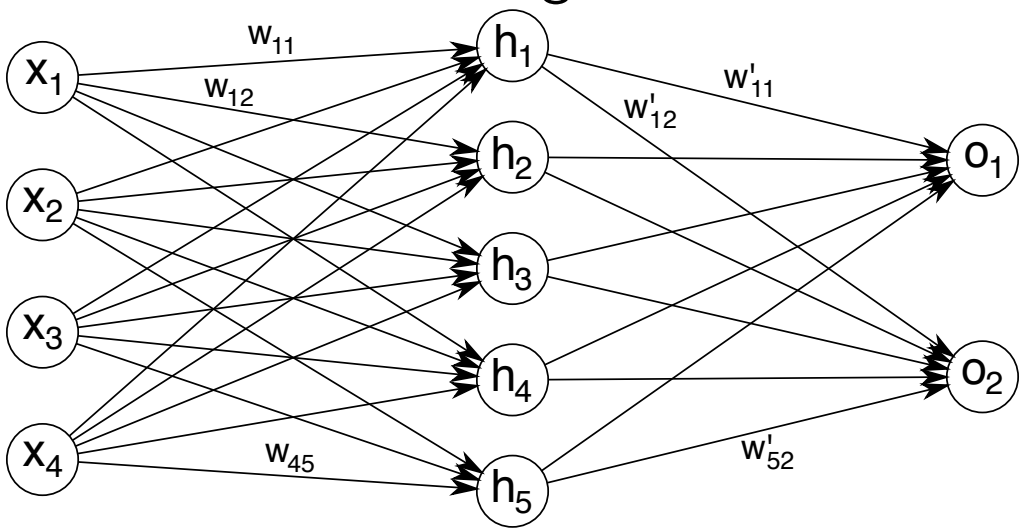


The error gradient is then

$$\frac{\partial}{\partial w} E = x\delta$$

Forward-Backward Algorithm

Consider a neural network with weights

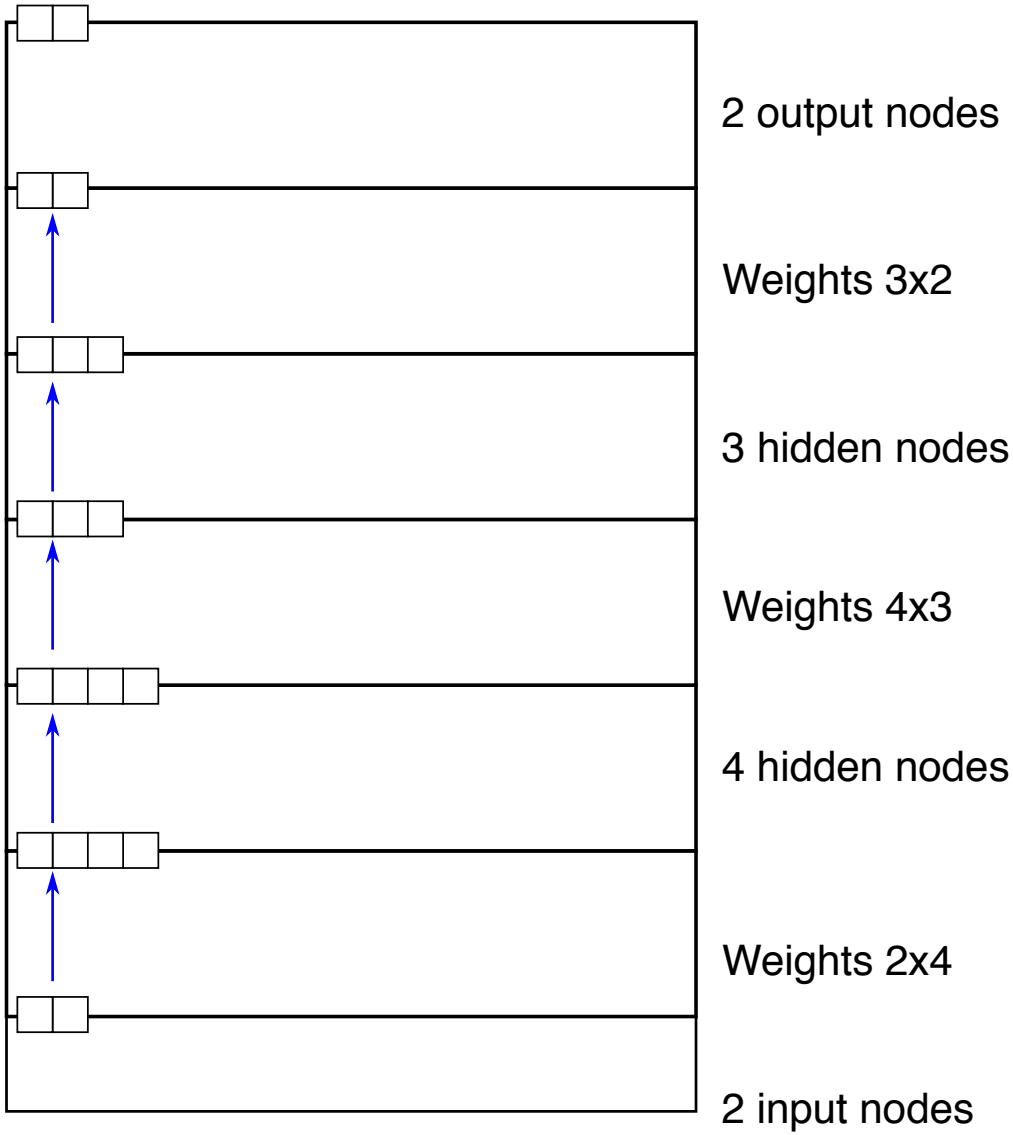


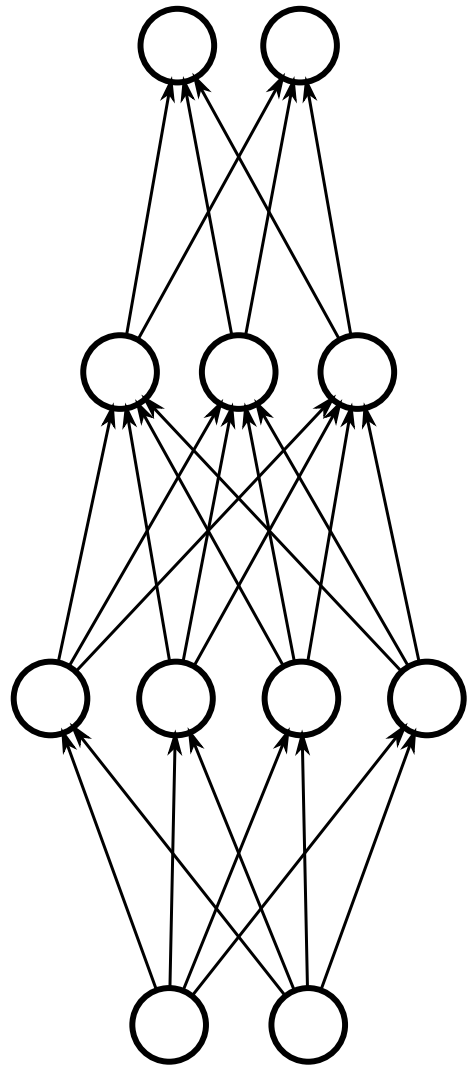
$$x \to Wx \to h(Wx) \to W'h(Wx) \to o\left(W'h(Wx)\right)$$

Forward-Backward Algorithm

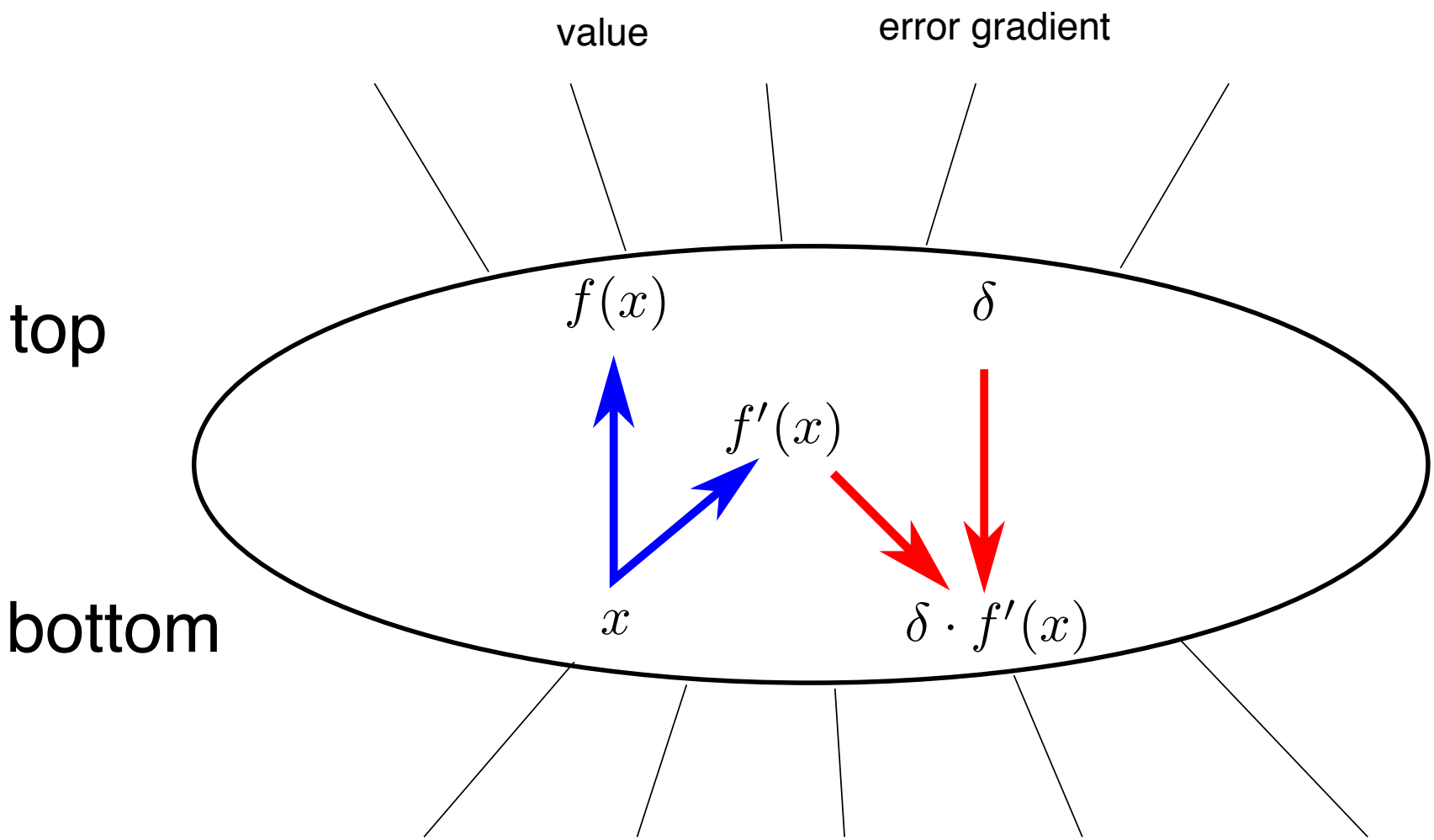
1. In the forward pass, we compute $NN(x)$, and with it, at each node $f(x)$ as well as $f'(x)$
2. To prepare the backward pass, we estimate the error gradient of the output given the target (i.e using cross-entropy, squared loss, etc.)
3. In the backward pass, we multiply the error gradient value along the path back to the input layer
4. The gradients of the weights are $x\delta$ for each weight that connects a node with output x to a node with error gradient δ
5. After some steps (one, few, or many) we update the weights

Forward-Backward Algorithm (in detail)





Forward-Backward Algorithm (Node)



forward pass

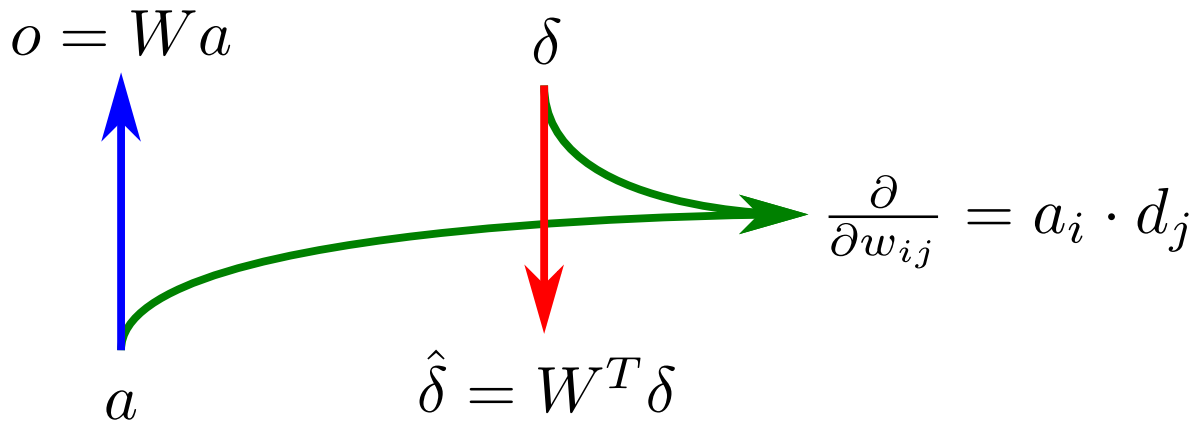
backward pass

Forward-Backward Algorithm (Weights)

value | error gradient | weight update

top

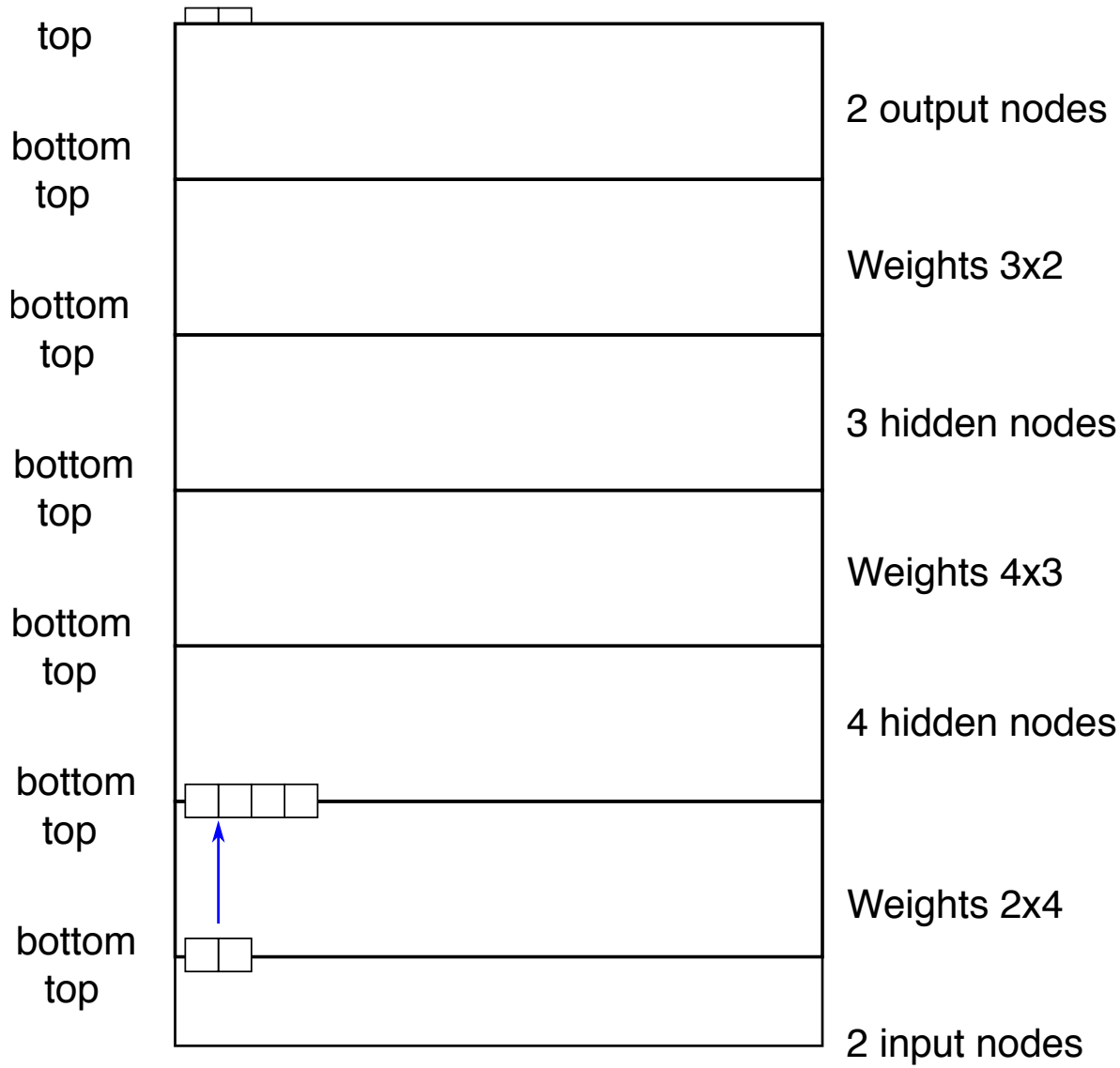
bottom

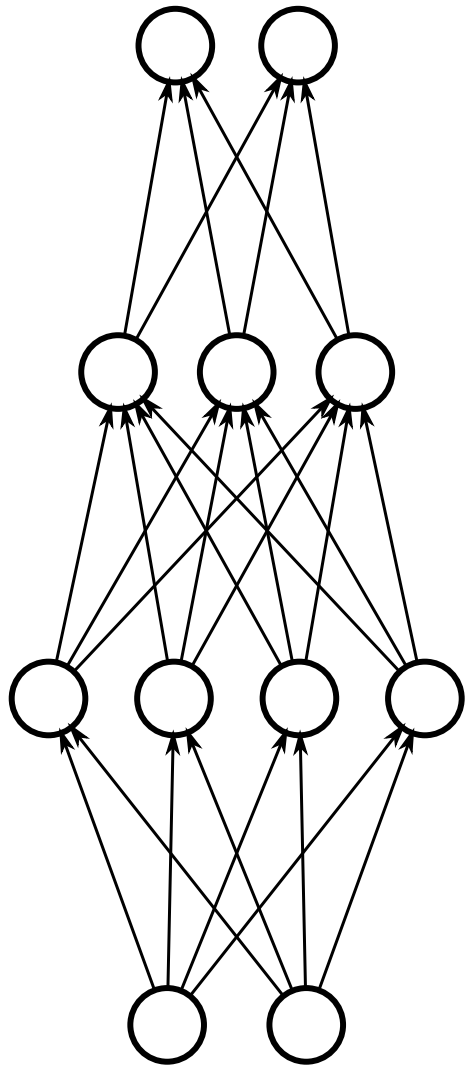


forward pass

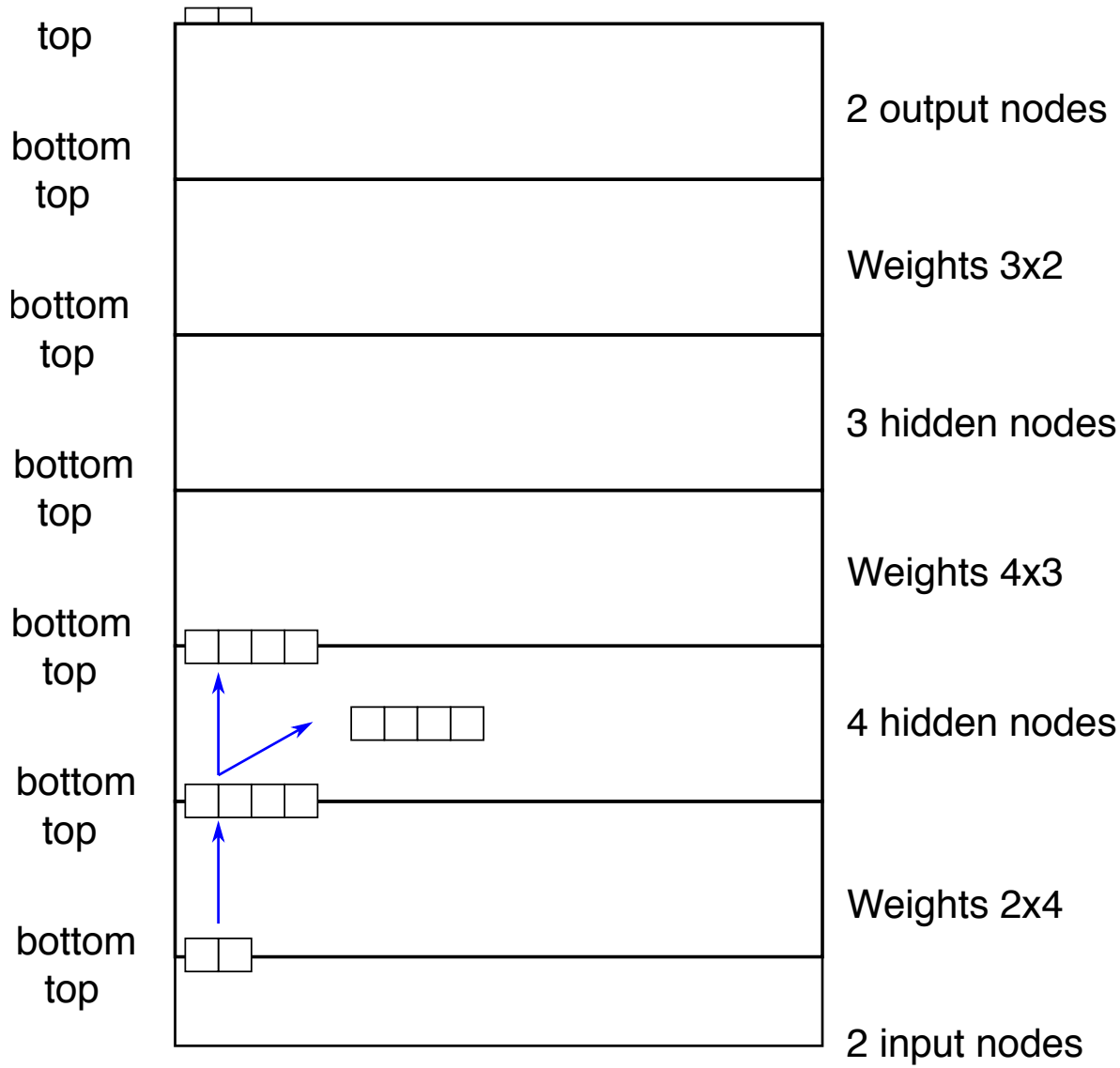
backward pass

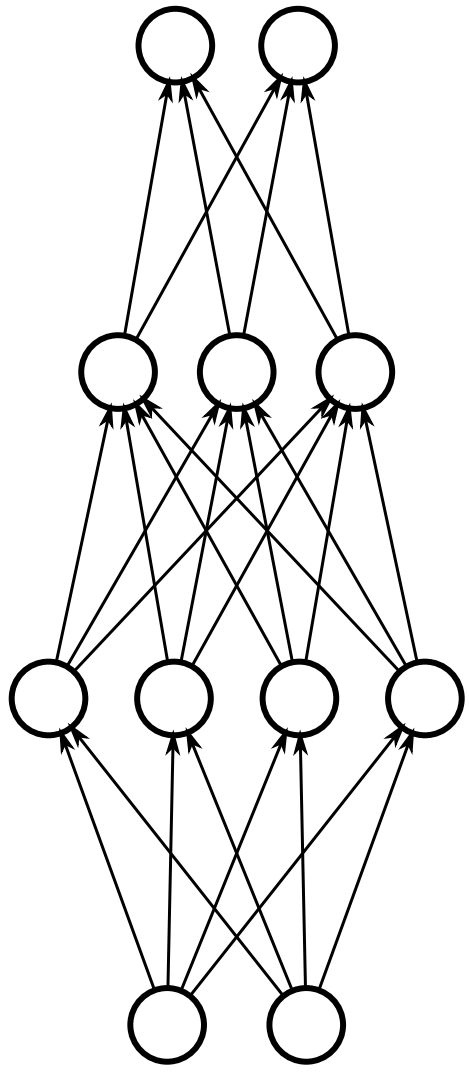
Forward Algorithm (All)



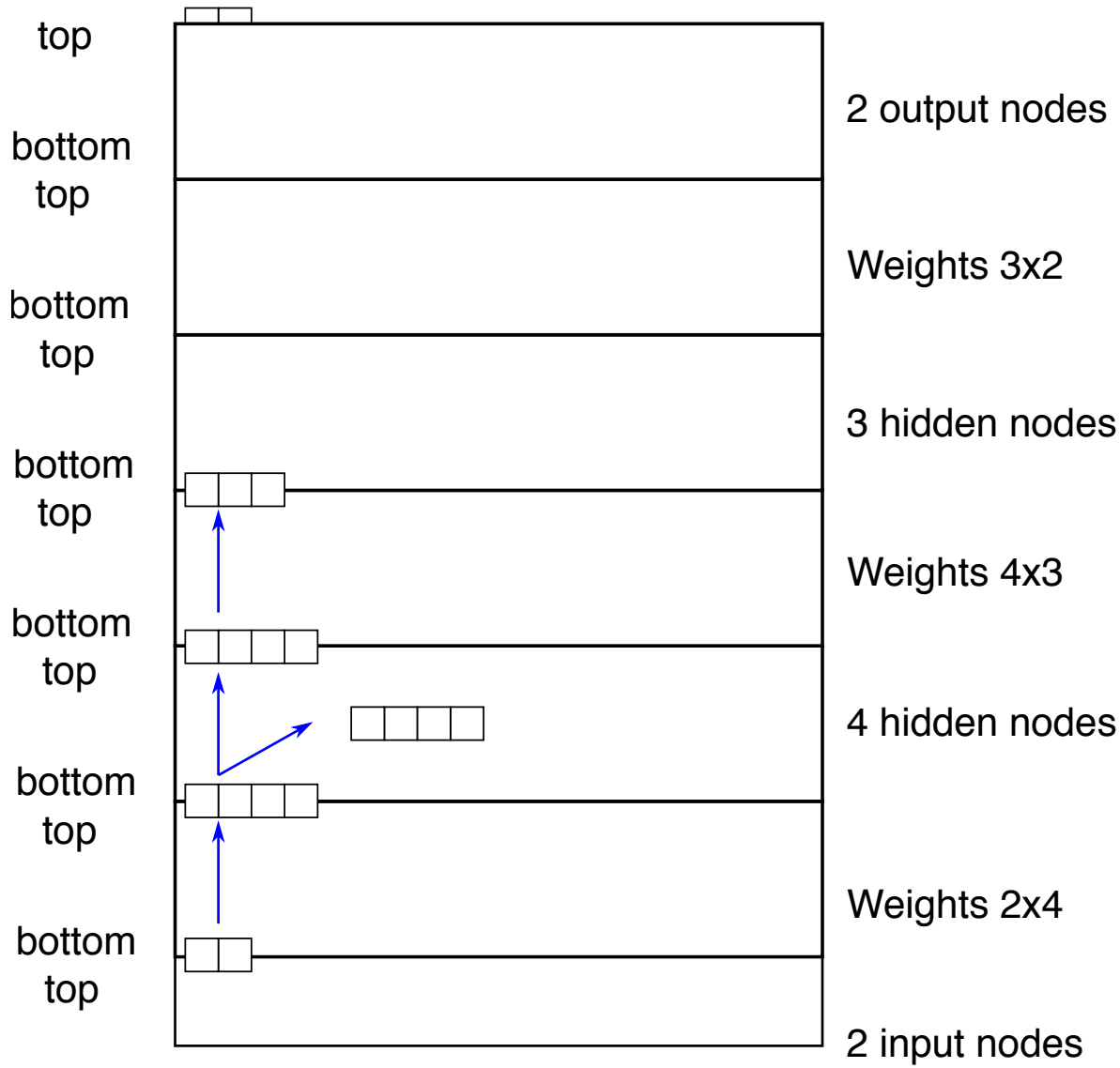


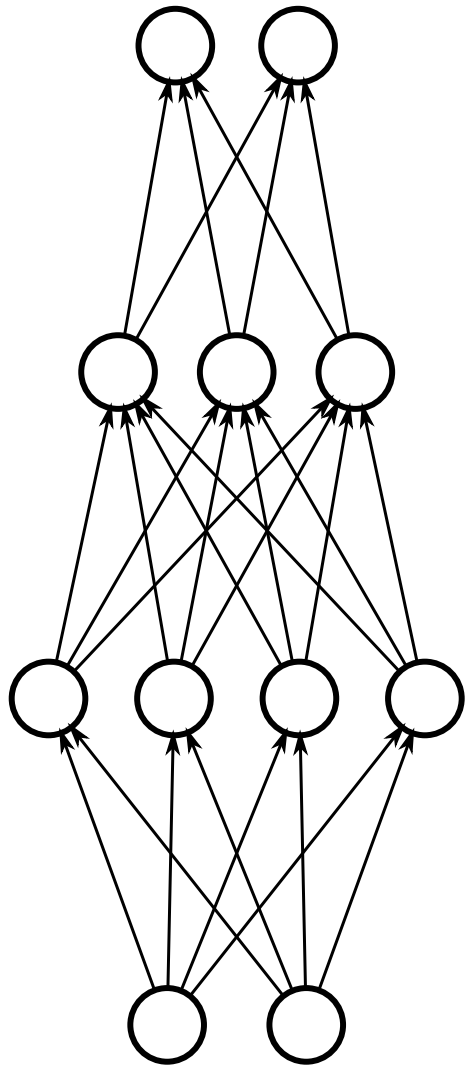
Forward Algorithm (All)



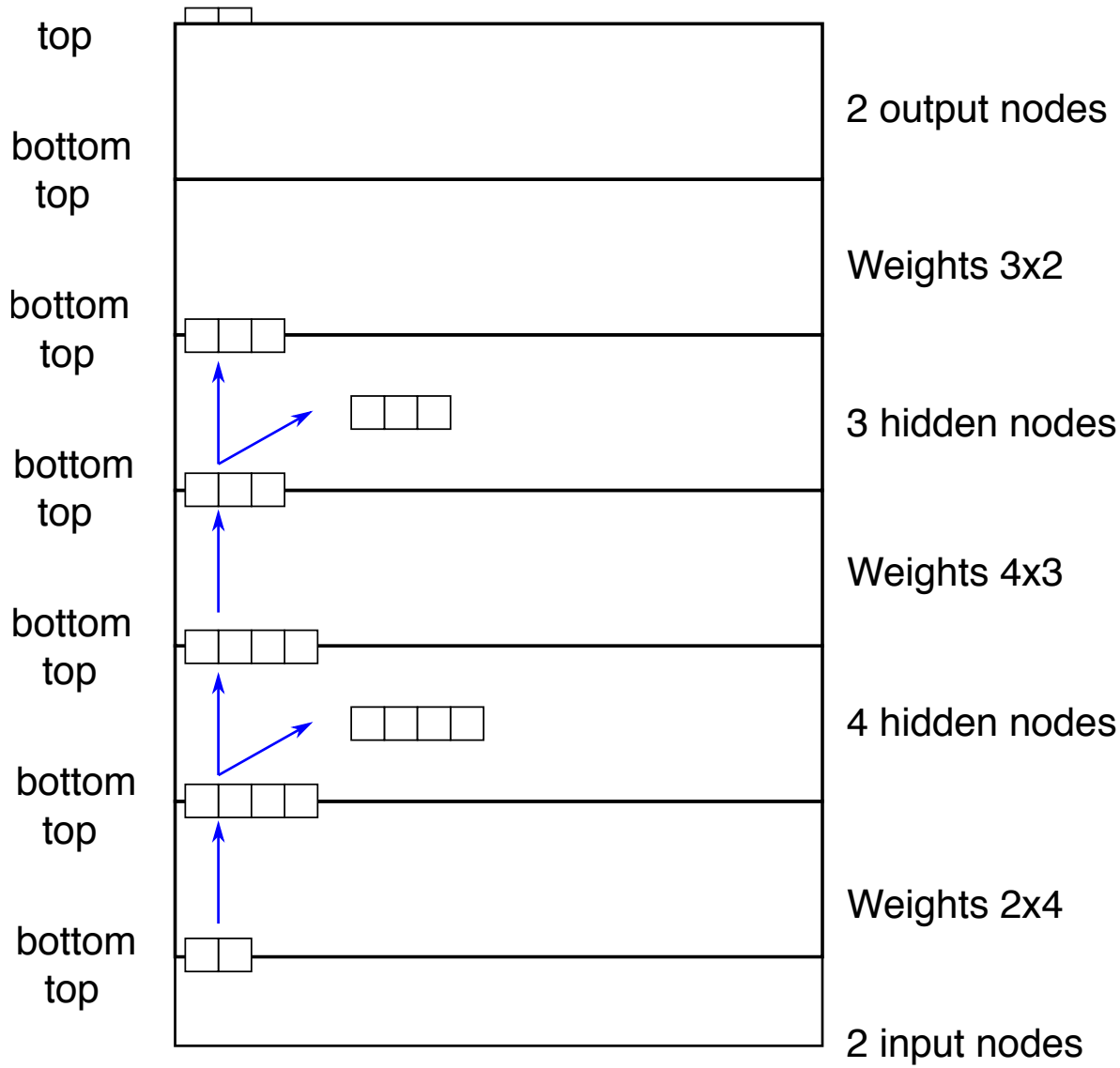


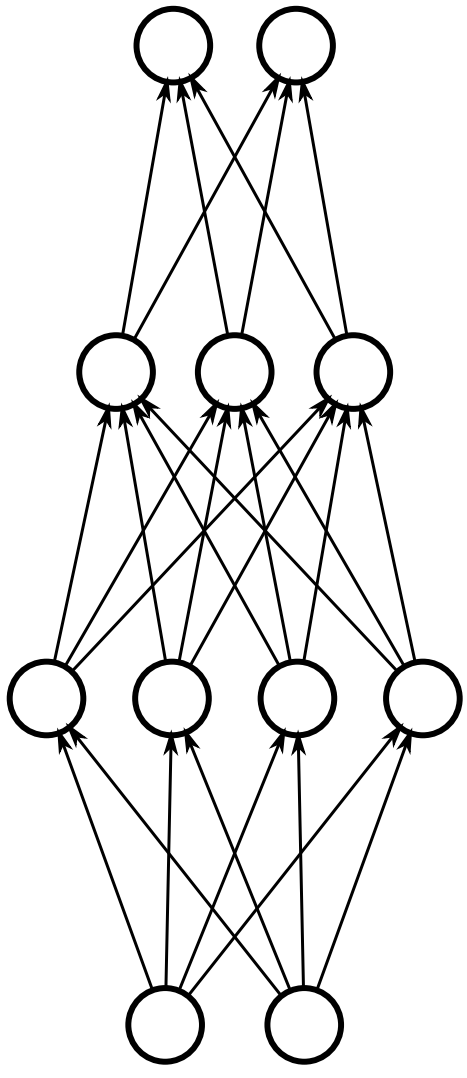
Forward Algorithm (All)



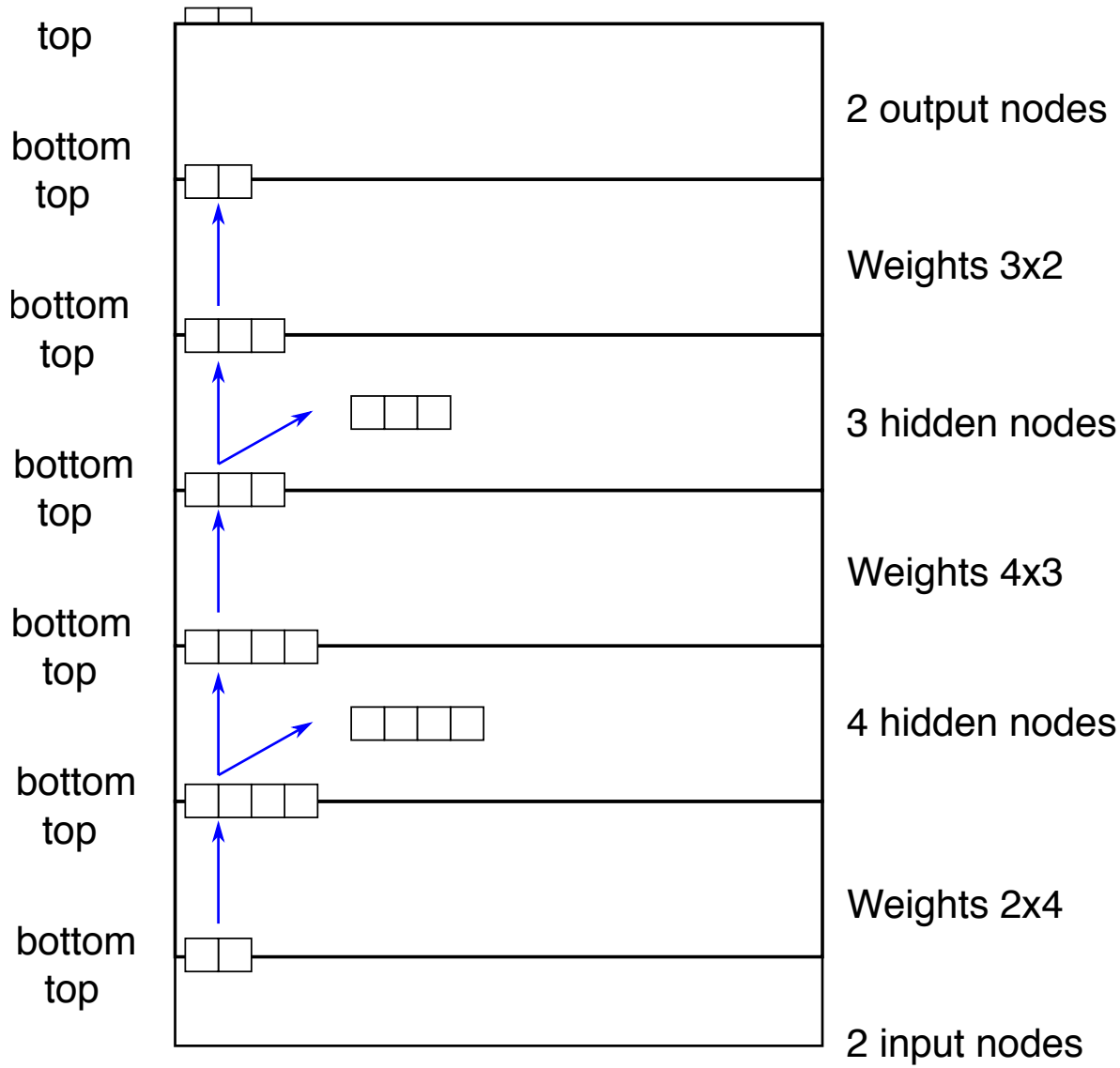


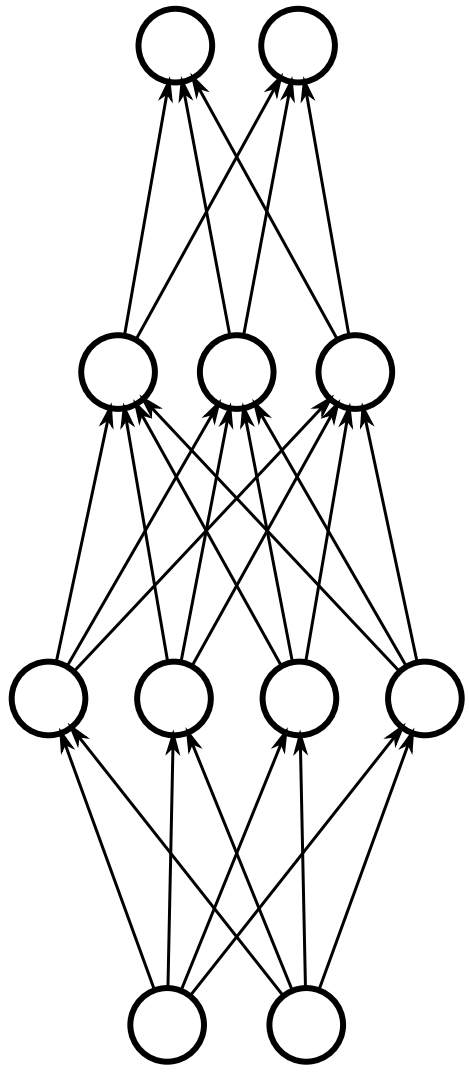
Forward Algorithm (All)



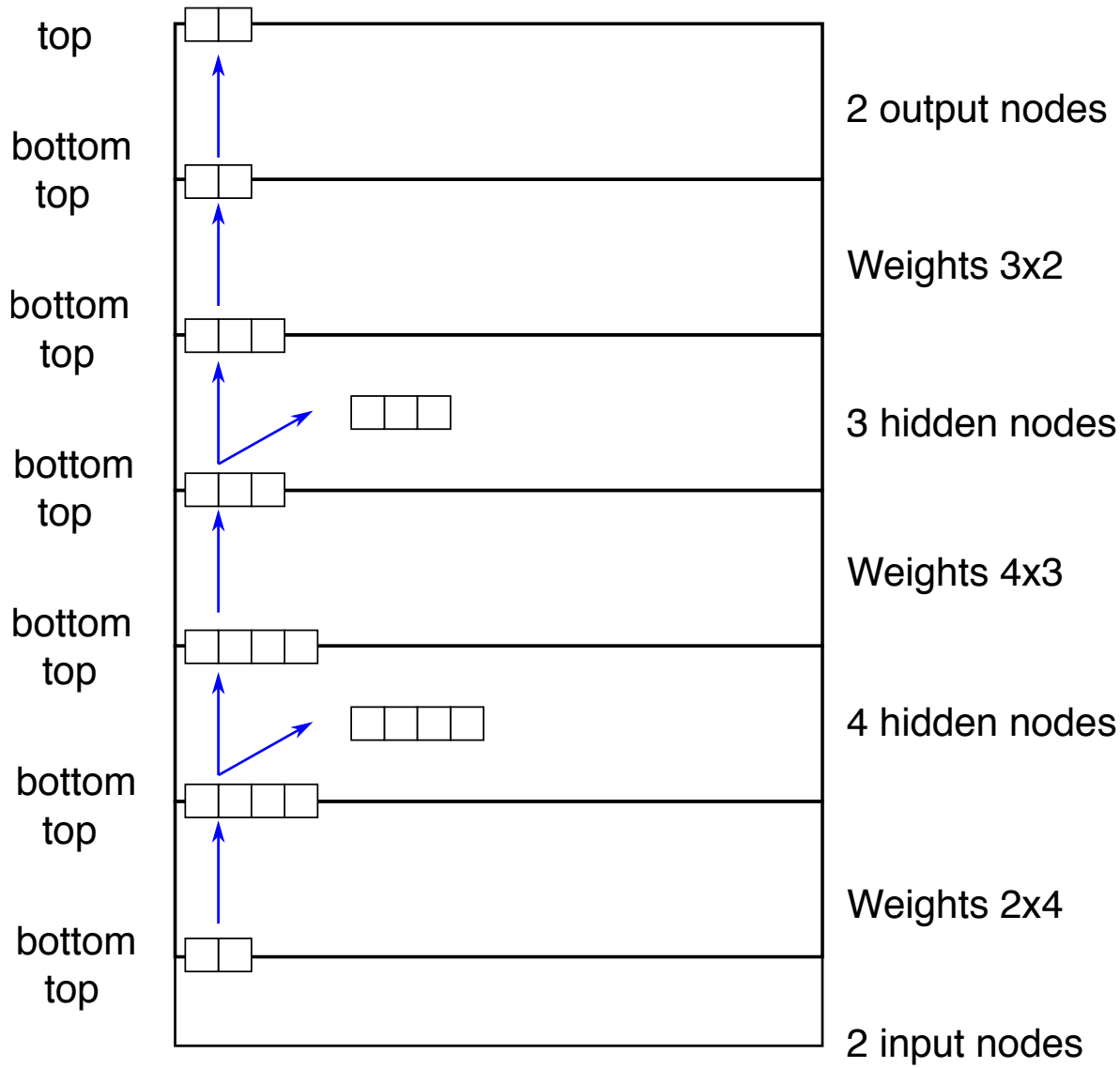


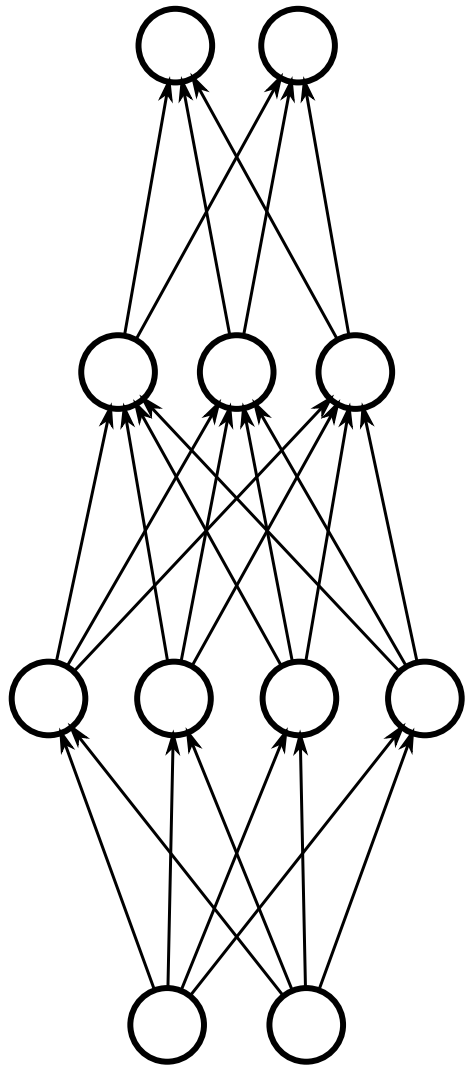
Forward Algorithm (All)



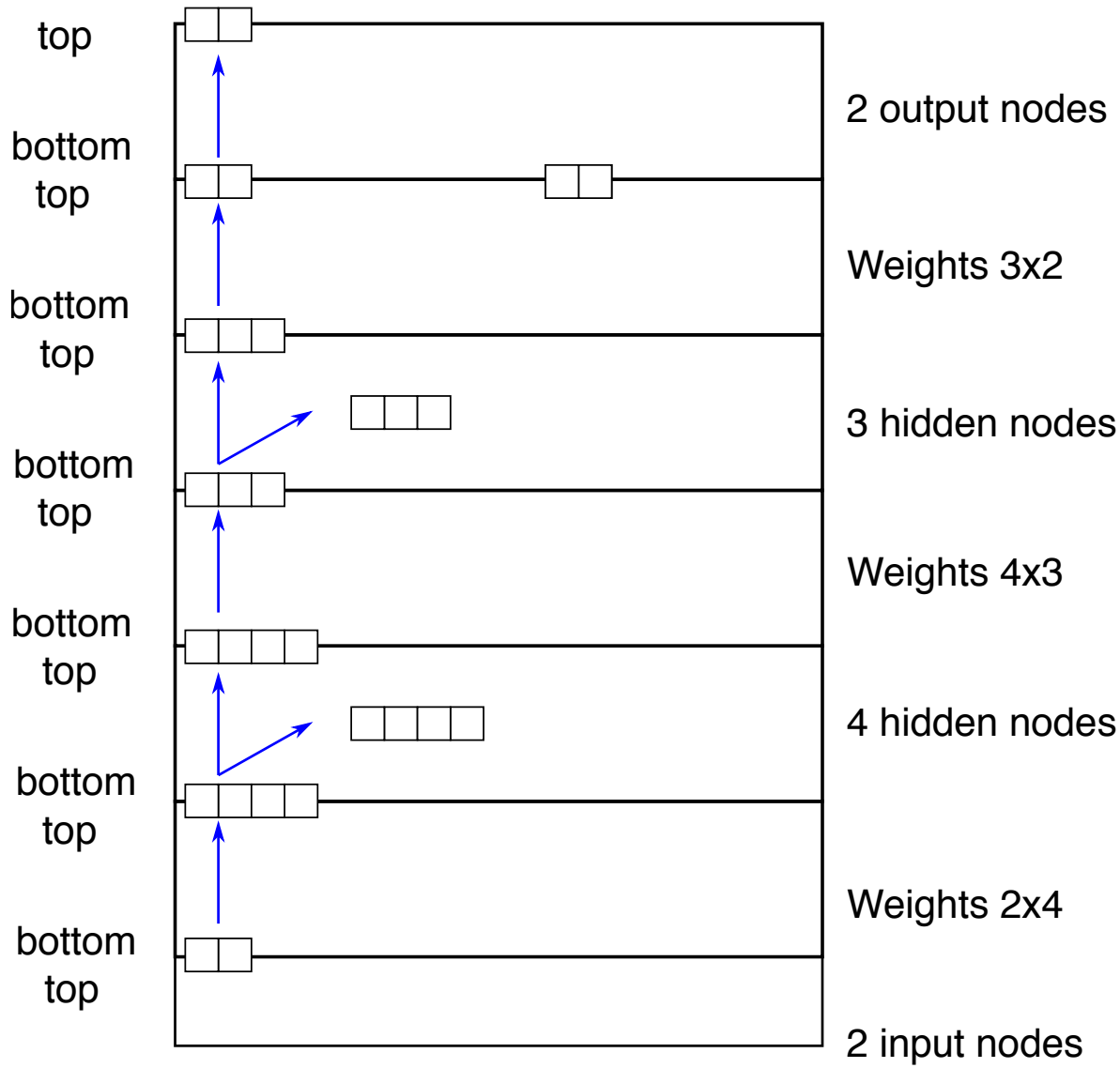


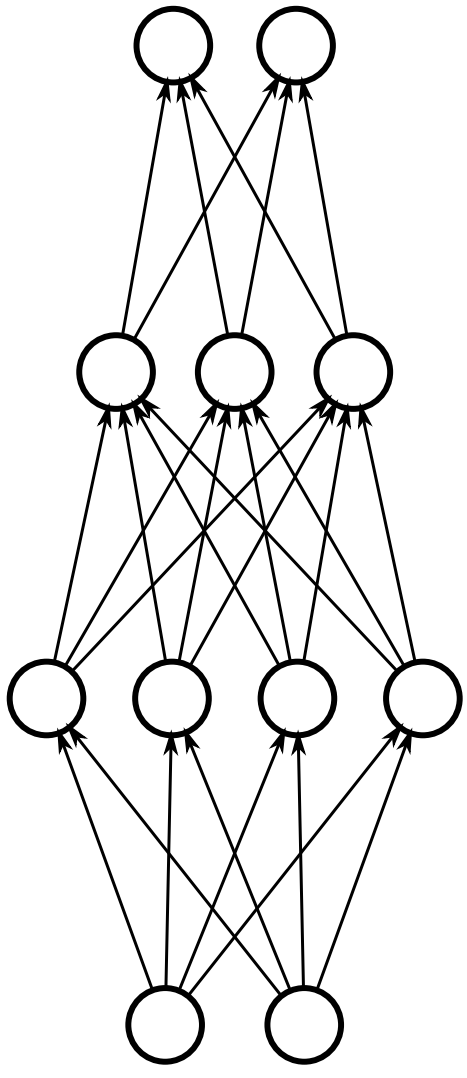
Forward Algorithm (All)



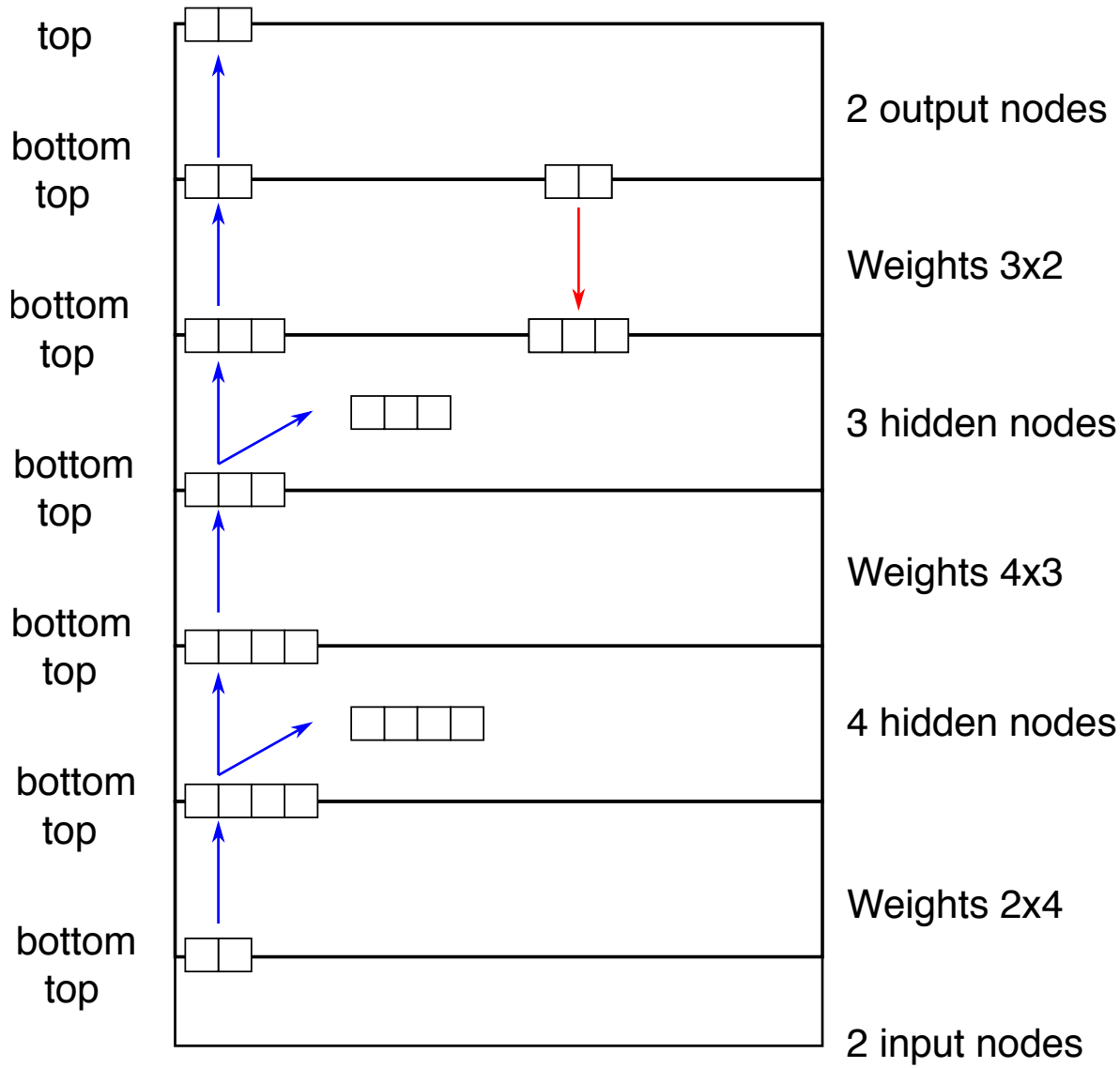


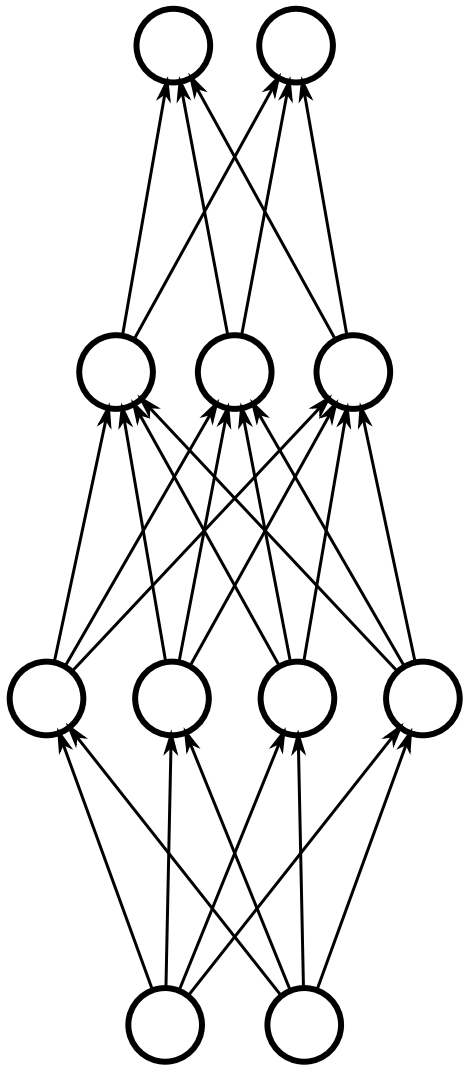
Backward Algorithm (All)



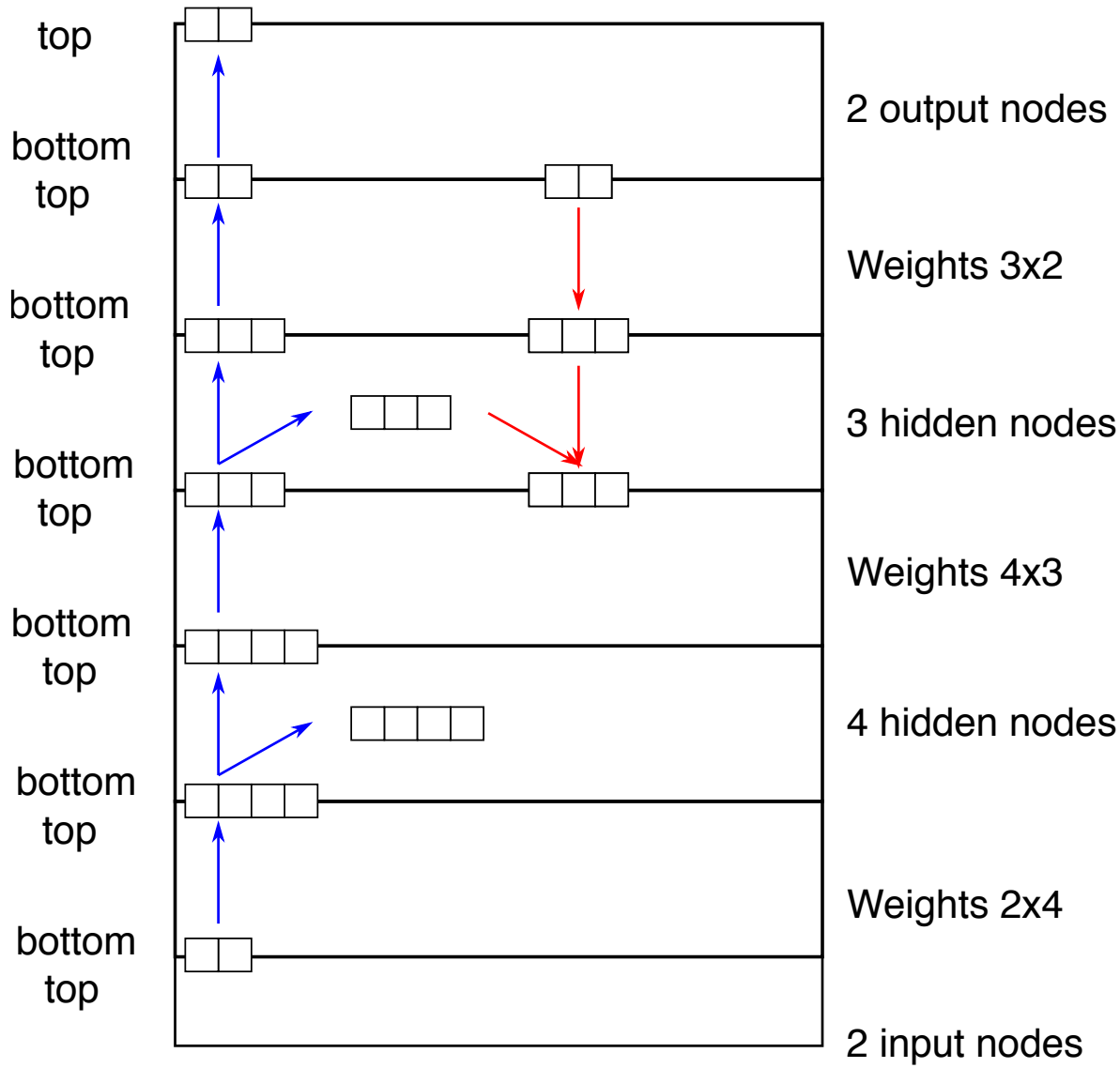


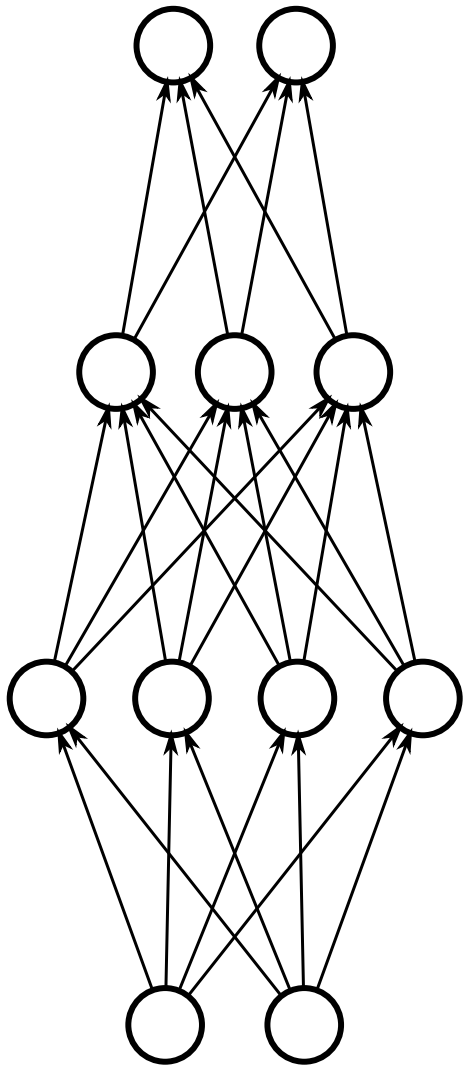
Backward Algorithm (All)



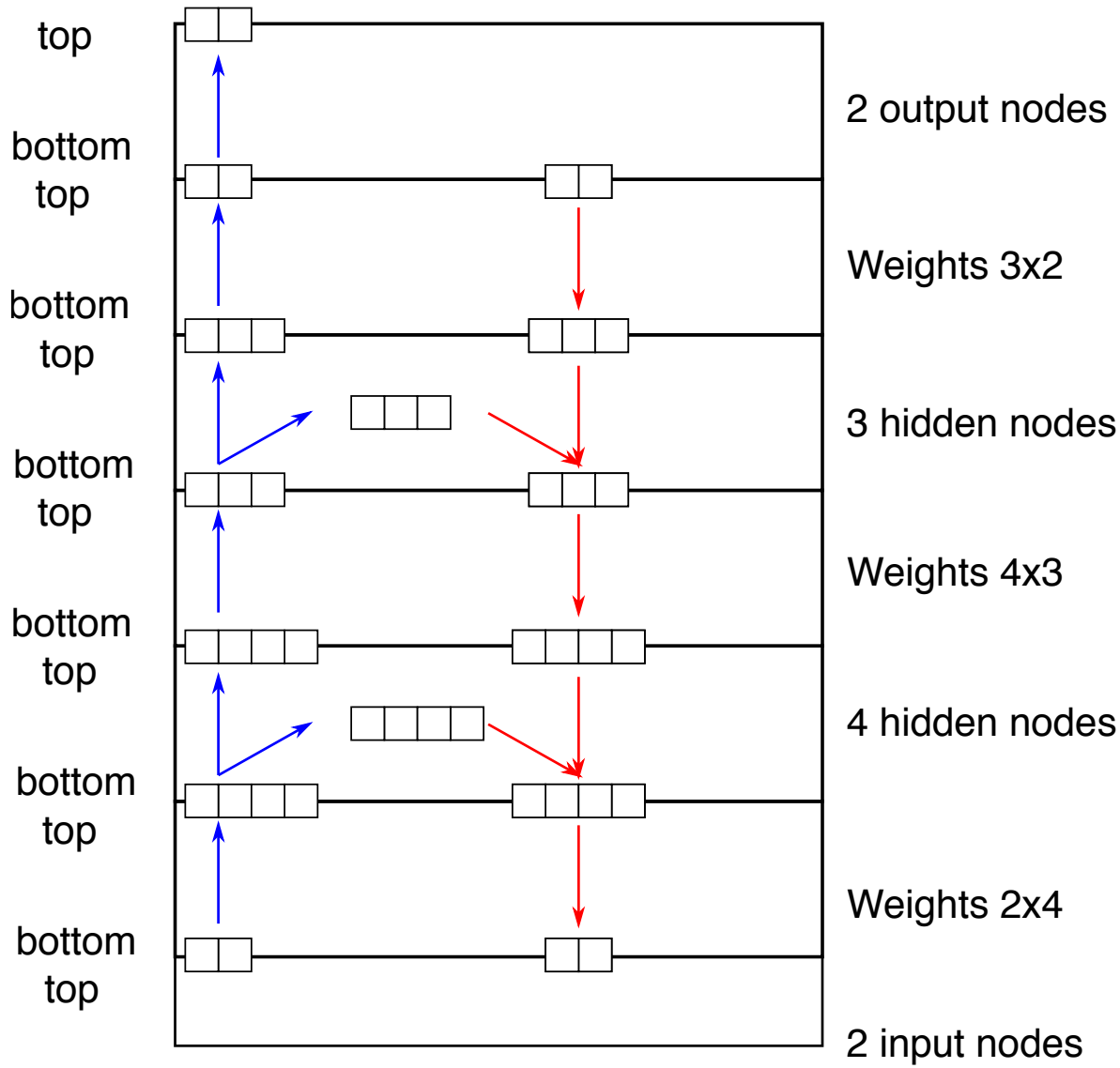


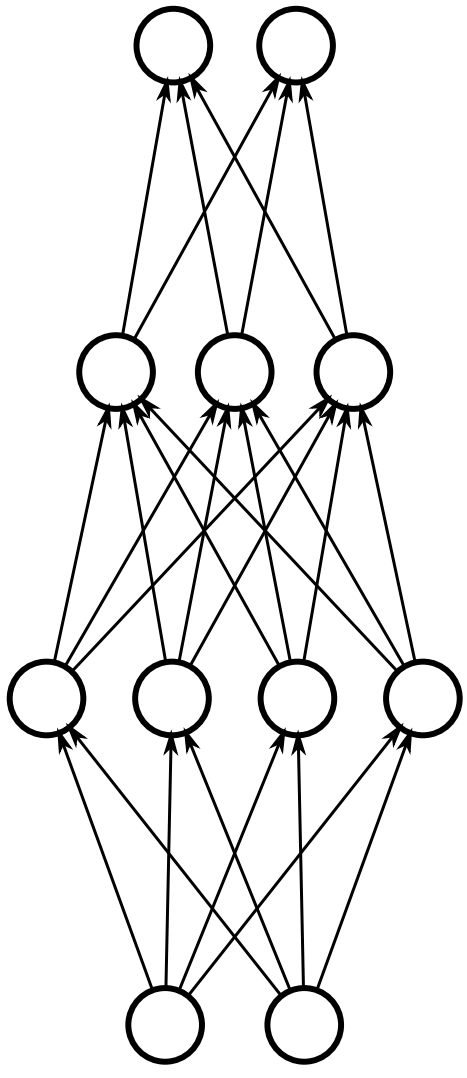
Backward Algorithm (All)



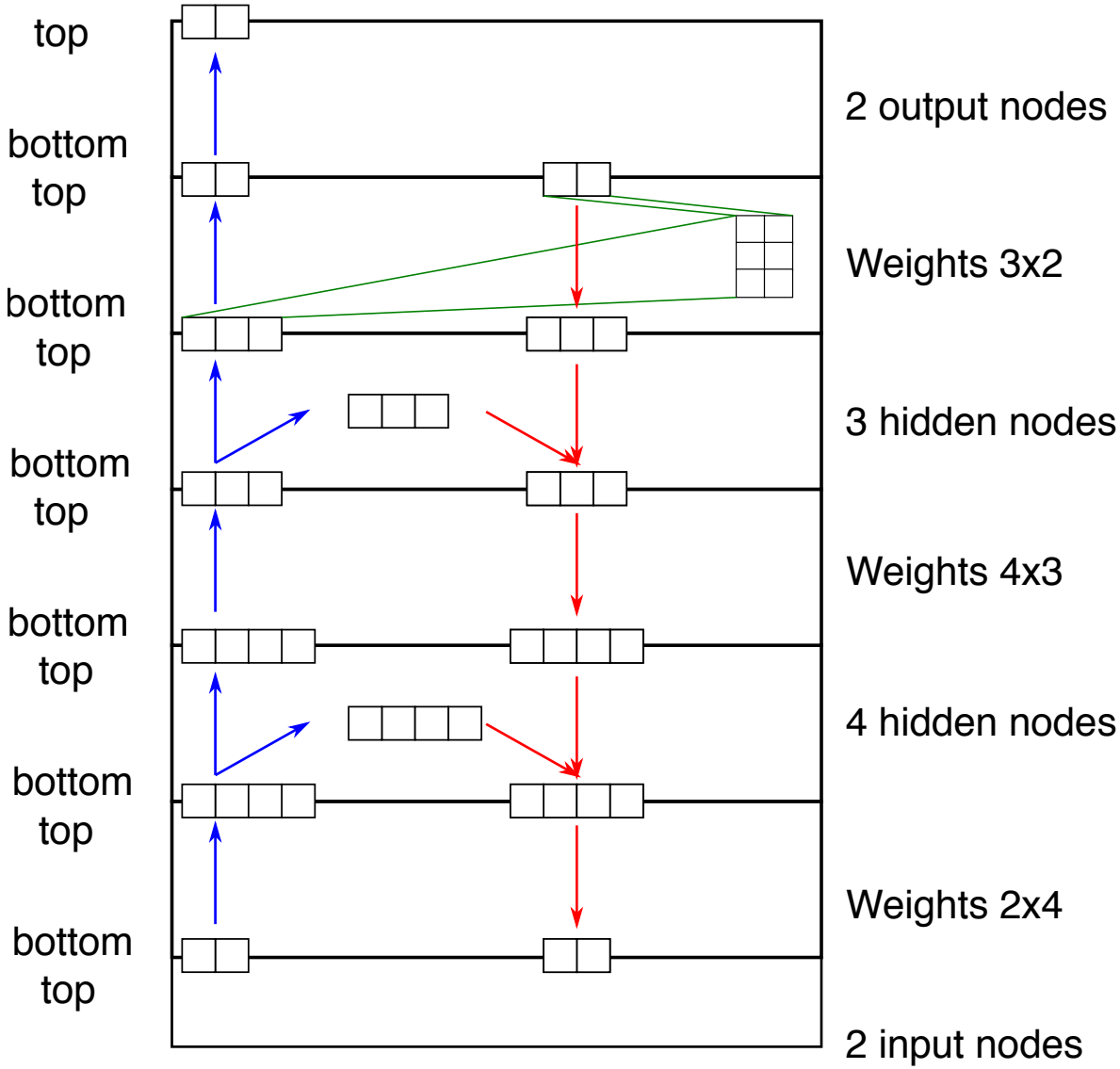


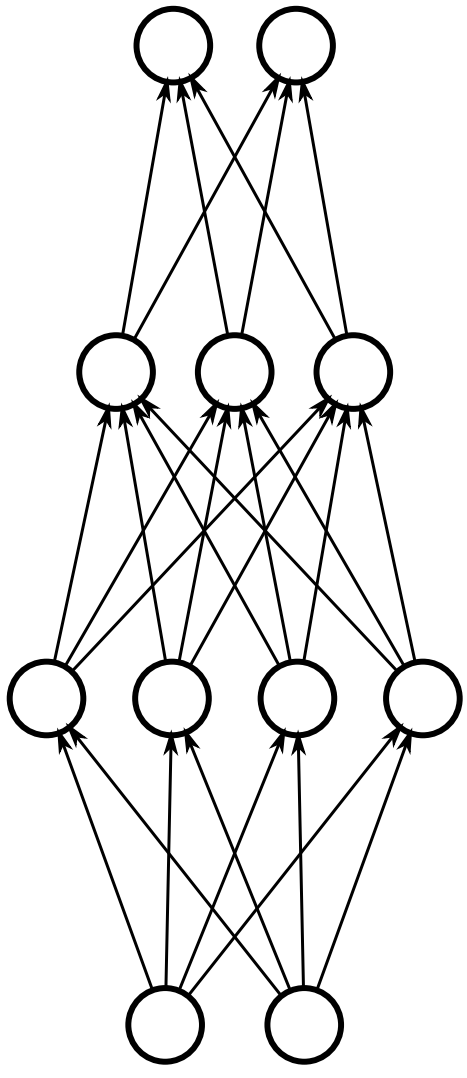
Backward Algorithm (All)





Weight Gradients Estimation





Weight Gradients Estimation