Deep Learning

Bojan Božić

TU Dublin

Summer 2021

Big Idea

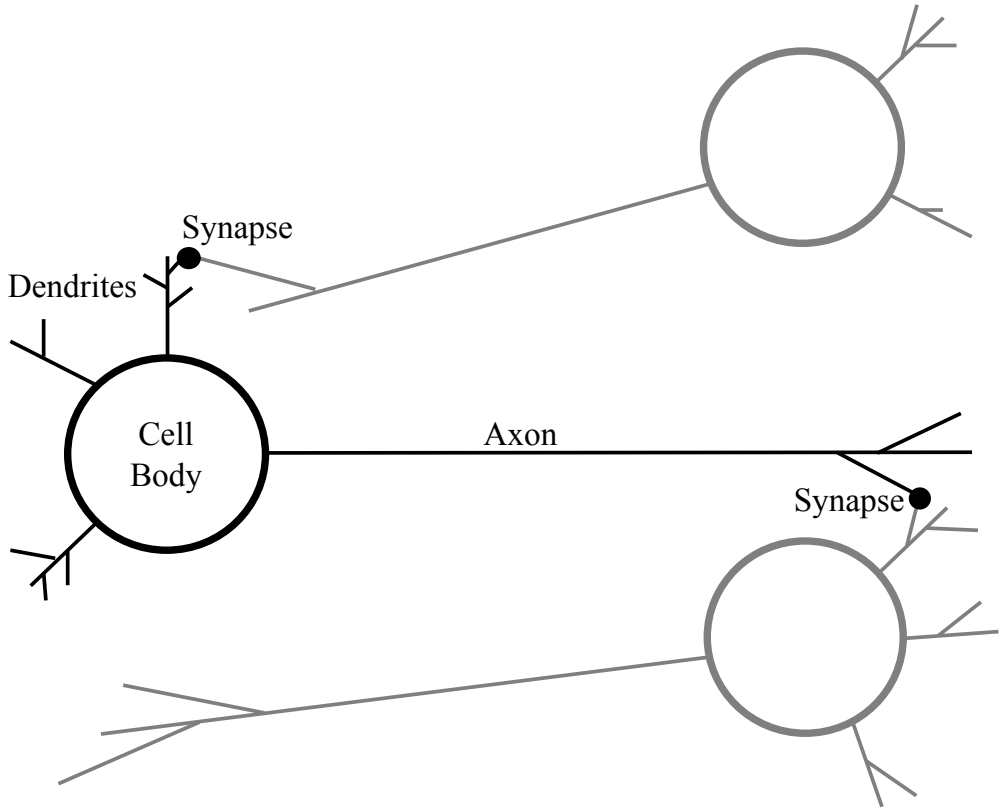


Figure 1: A high-level schematic of the structure of a neuron. This figure illustrates three interconnected neurons; the middle neuron is highlighted in black, and the major structural components of this neuron are labeled cell body, dendrites, and axon. Also marked are the synapses connecting the axon of one neuron and the dendrite of another, which allow signals to pass between the neurons.

Fundamentals

$$z = \underbrace{\mathbf{w}[0] \times \mathbf{d}[0] + \mathbf{w}[1] \times \mathbf{d}[1] + \cdots + \mathbf{w}[m] \times \mathbf{d}[m]}_{weighted\ sum} \tag{1}$$

$$= \sum_{j=0}^{m} \mathbf{w}[j] \times \mathbf{d}[j]$$

$$= \underbrace{\mathbf{w} \cdot \mathbf{d}}_{dot\ product} = \underbrace{\mathbf{w}^T \mathbf{d}}_{matrix\ product} = [w_0, w_1, \ldots, w_m] \begin{bmatrix} d_0 \\ d_2 \\ \vdots \\ d_m \end{bmatrix} \tag{2}$$

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } z \geq \theta \\ 0 & otherwise \end{cases} \tag{3}$$

$$rectifier(z) = max(0, z) \tag{4}$$

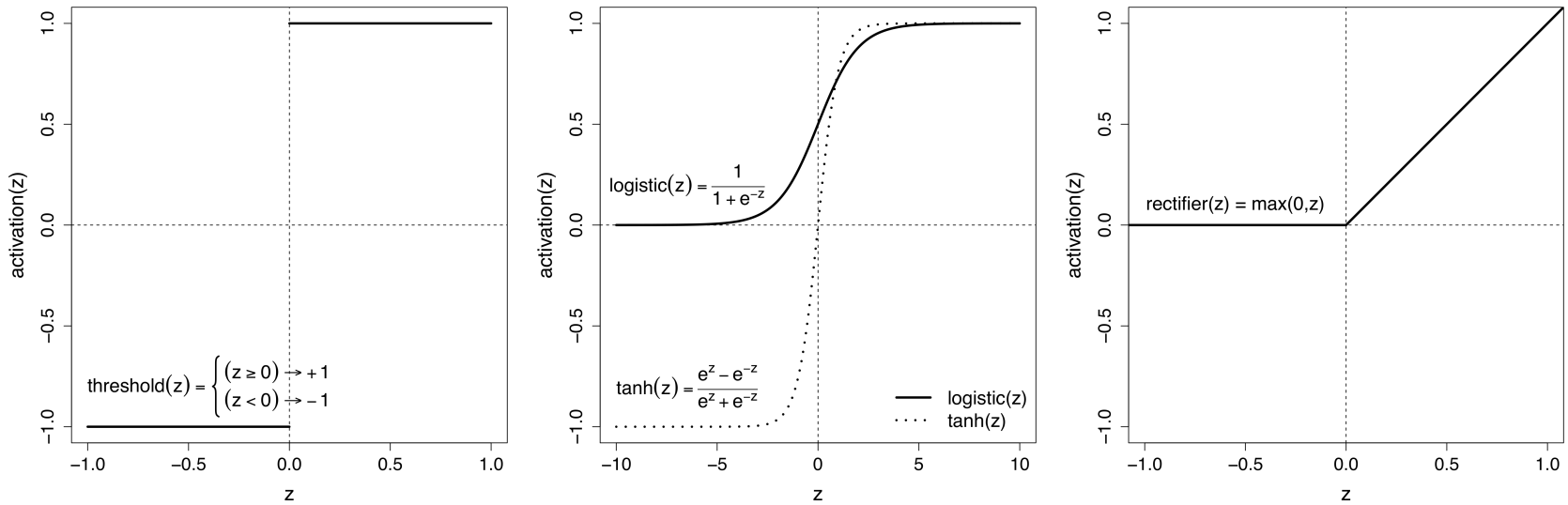


Figure 2: Plots for activation functions that have been popular in the history of neural networks.

$$\begin{aligned}
\mathbb{M}_{\mathbf{w}}(\mathbf{d}) &= \varphi\left(\mathbf{w}\left[0\right] \times \mathbf{d}\left[0\right] + \mathbf{w}\left[1\right] \times \mathbf{d}\left[1\right] + \cdots + \mathbf{w}\left[m\right] \times \mathbf{d}\left[m\right]\right) \\
&= \varphi\left(\sum_{i=0}^{m} w_i \times d_i\right) = \varphi\left(\underbrace{\mathbf{w} \cdot \mathbf{d}}_{dot\ product}\right) \\
&= \varphi\left(\underbrace{\mathbf{w}^T \mathbf{d}}_{matrix\ product}\right) = \varphi\left([w_0, w_1, \ldots, w_m] \begin{bmatrix} d_0 \\ d_2 \\ \vdots \\ d_m \end{bmatrix}\right)
\end{aligned} \tag{5}$$

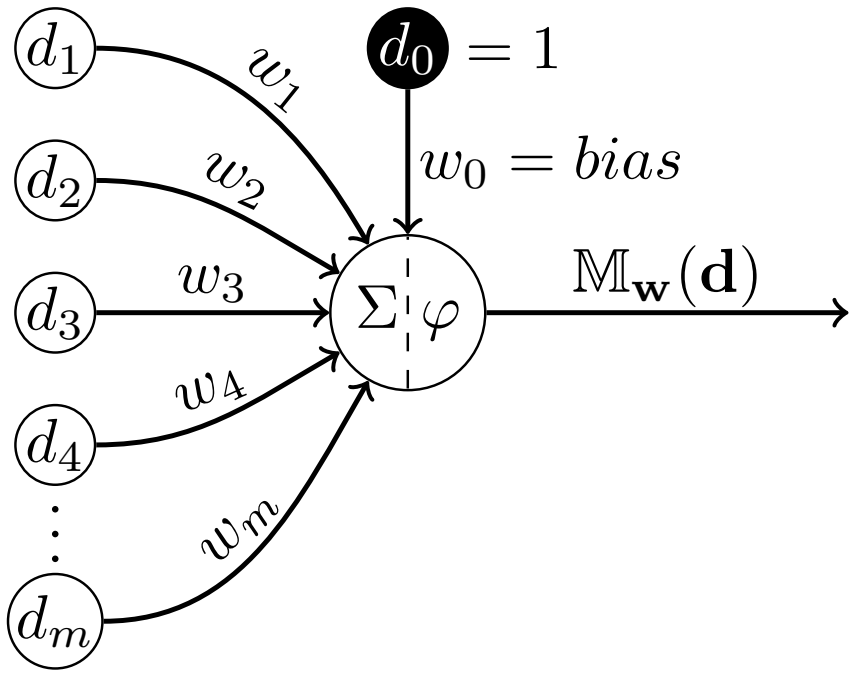


Figure 3: A schematic of an artificial neuron.

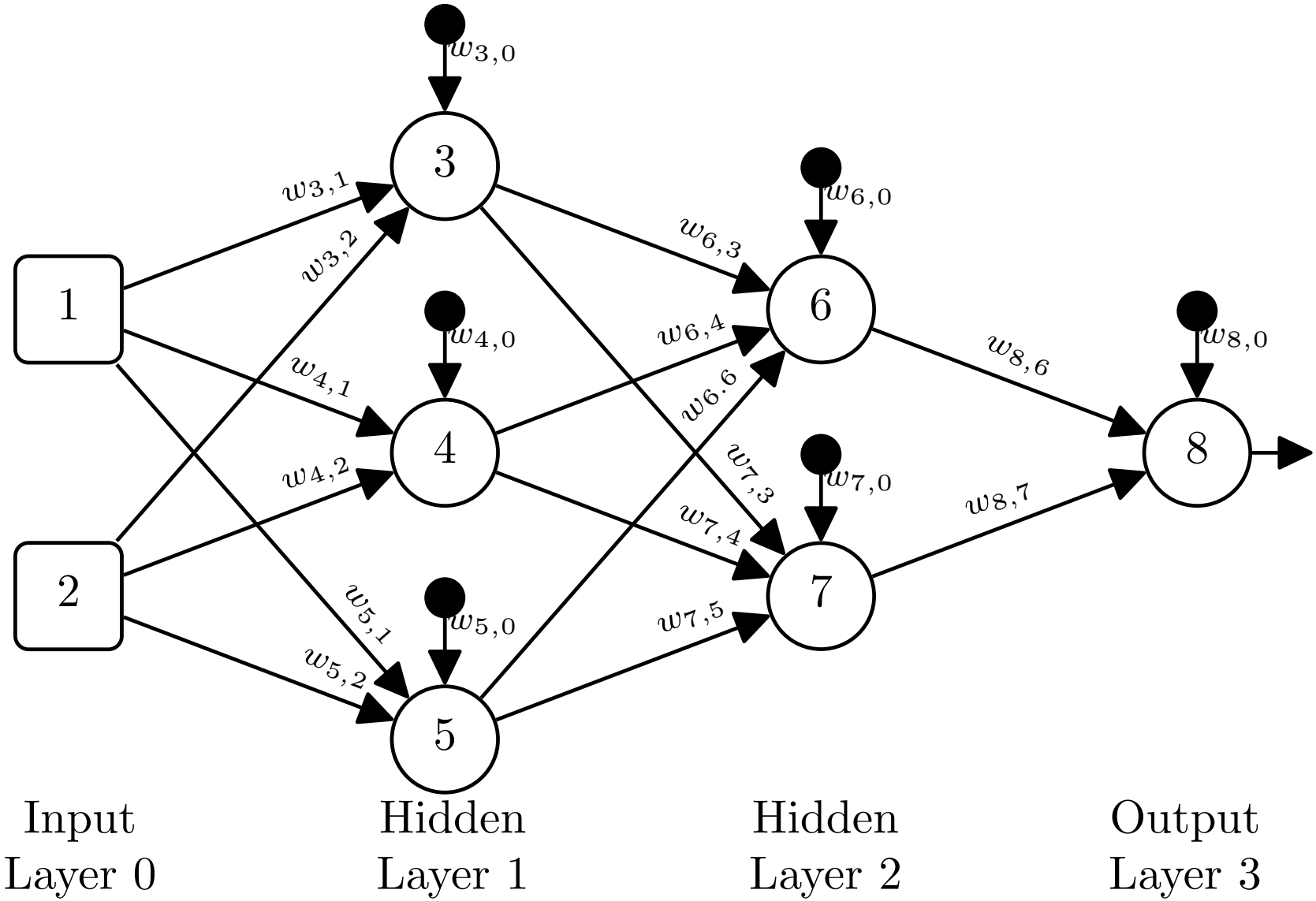


Figure 4: A schematic of a feedforward artificial neural network.

$$\mathbf{z}^{(\mathbf{2})} = \mathbf{W}^{(\mathbf{2})}\mathbf{a}^{(\mathbf{1})} \tag{6}$$

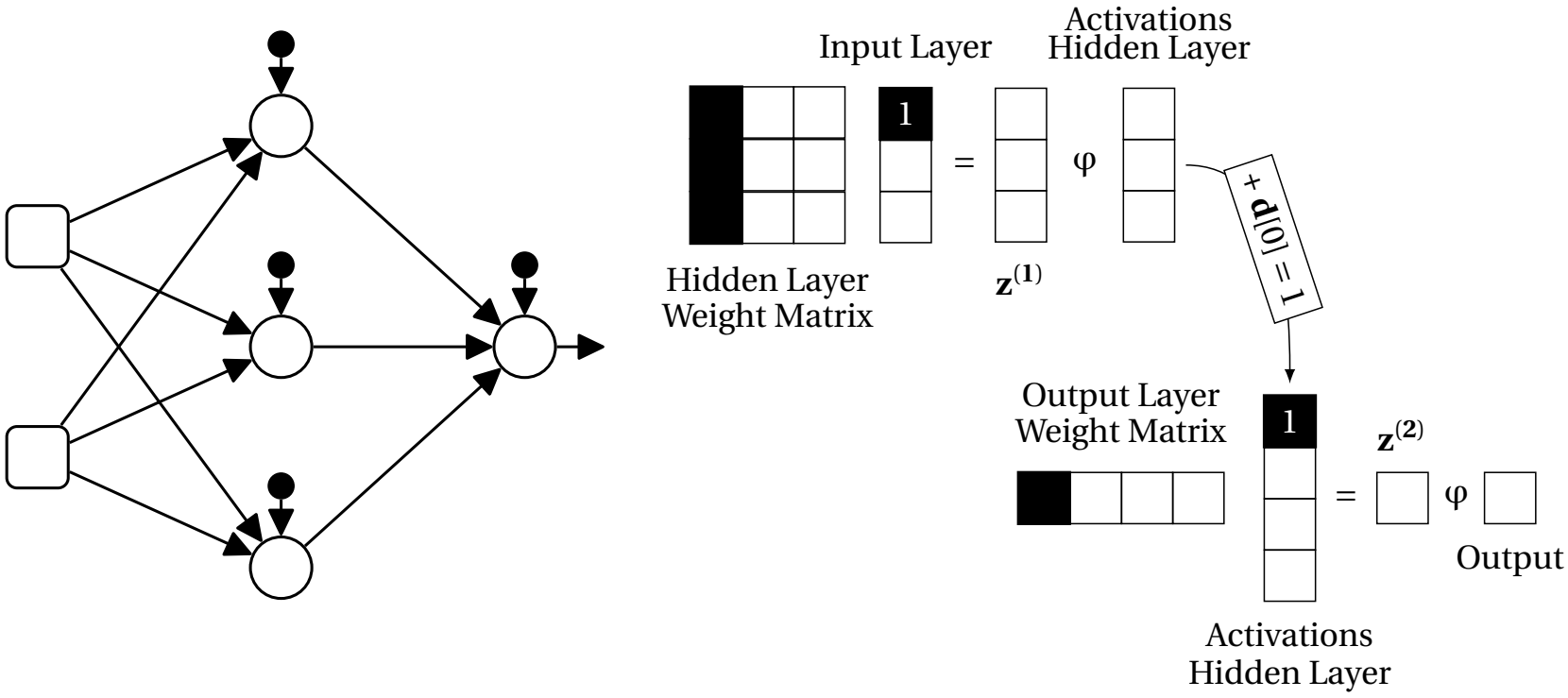


Figure 5: An illustration of the correspondence between graphical and matrix representations of a neural network. This figure is inspired by Figure 3.9 of (Kelleher, 2019).

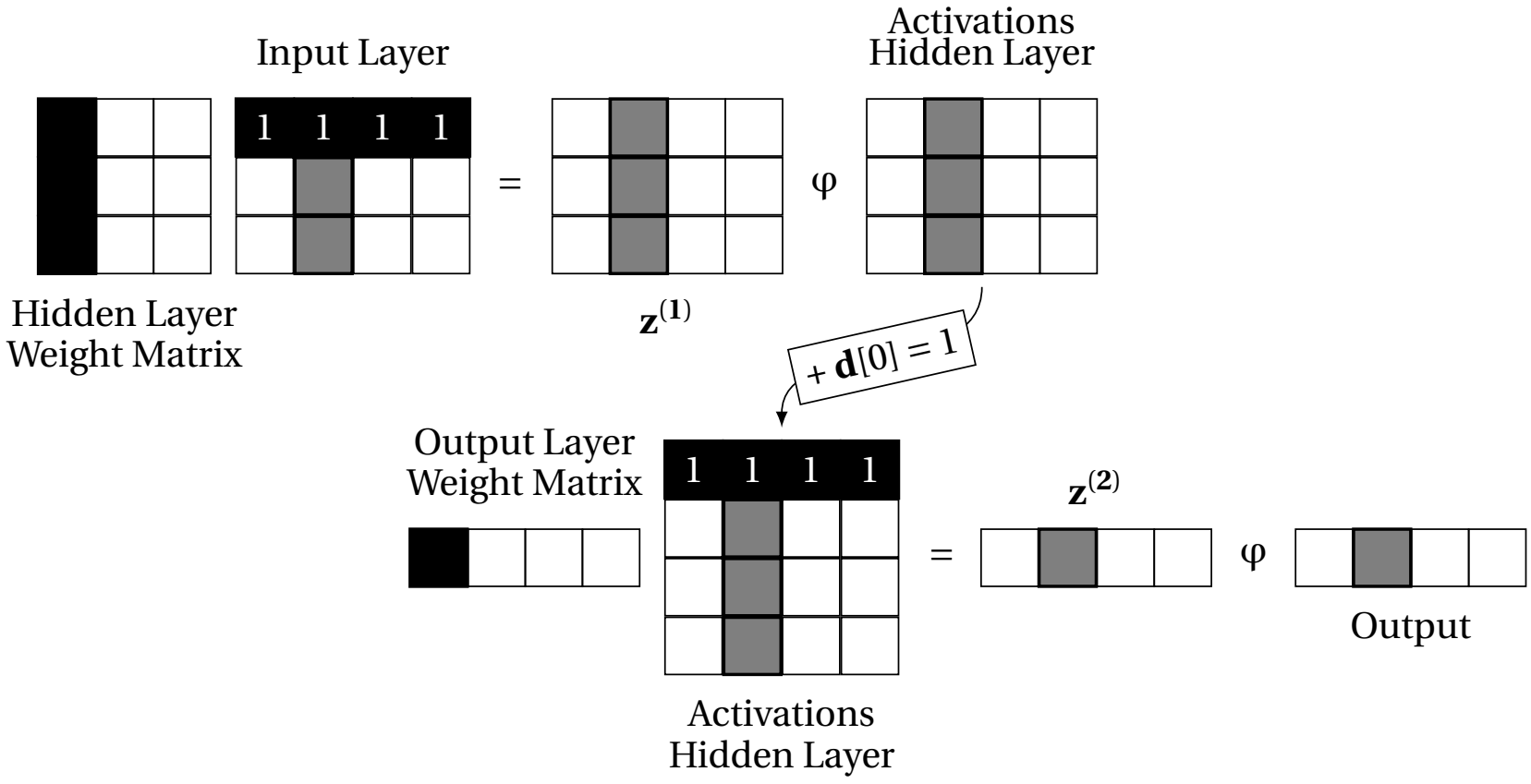


Figure 6: An illustration of how a batch of examples can be processed in parallel using matrix operations.

$$\mathbf{A}^{(1)} = \mathbf{W}^{(1)}\mathbf{A}^{(0)} \tag{7}$$

$$\mathbf{A}^{(2)} = \mathbf{W}^{(2)}\mathbf{A}^{(1)} \tag{8}$$

$$\mathbf{A}^{(2)} = \mathbf{W}^{(2)}\left(\mathbf{W}^{(1)}\mathbf{A}^{(0)}\right) \tag{9}$$

$$\mathbf{A}^{(2)} = \left(\mathbf{W}^{(2)}\mathbf{W}^{(1)}\right)\mathbf{A}^{(0)} \tag{10}$$

$$\mathbf{A}^{(2)} = \mathbf{W}'\mathbf{A}^{(0)} \tag{11}$$

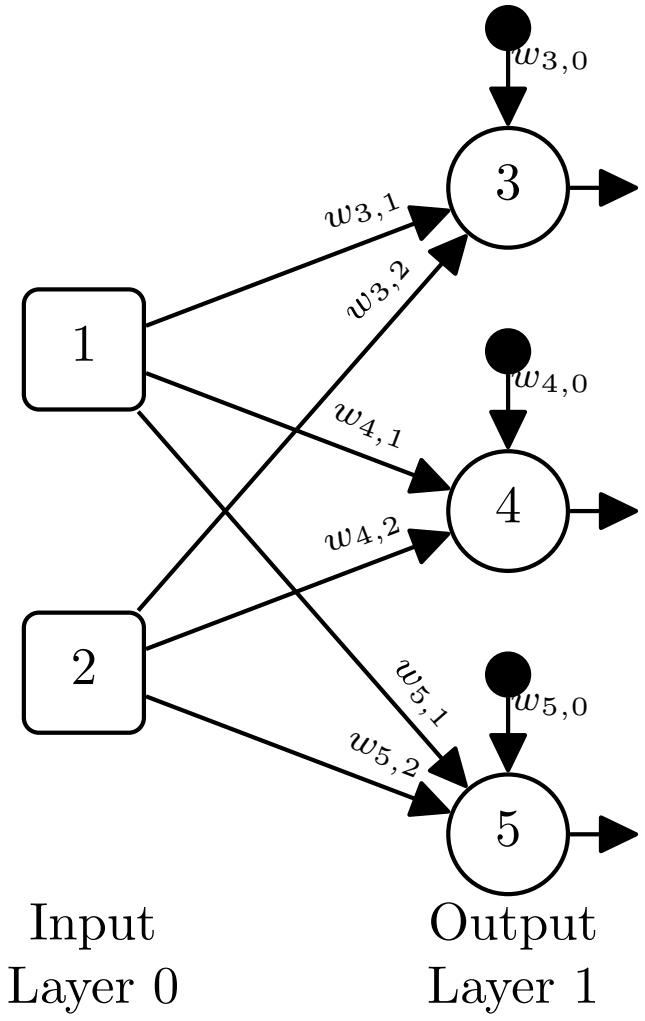


Figure 7: A single-layer network.

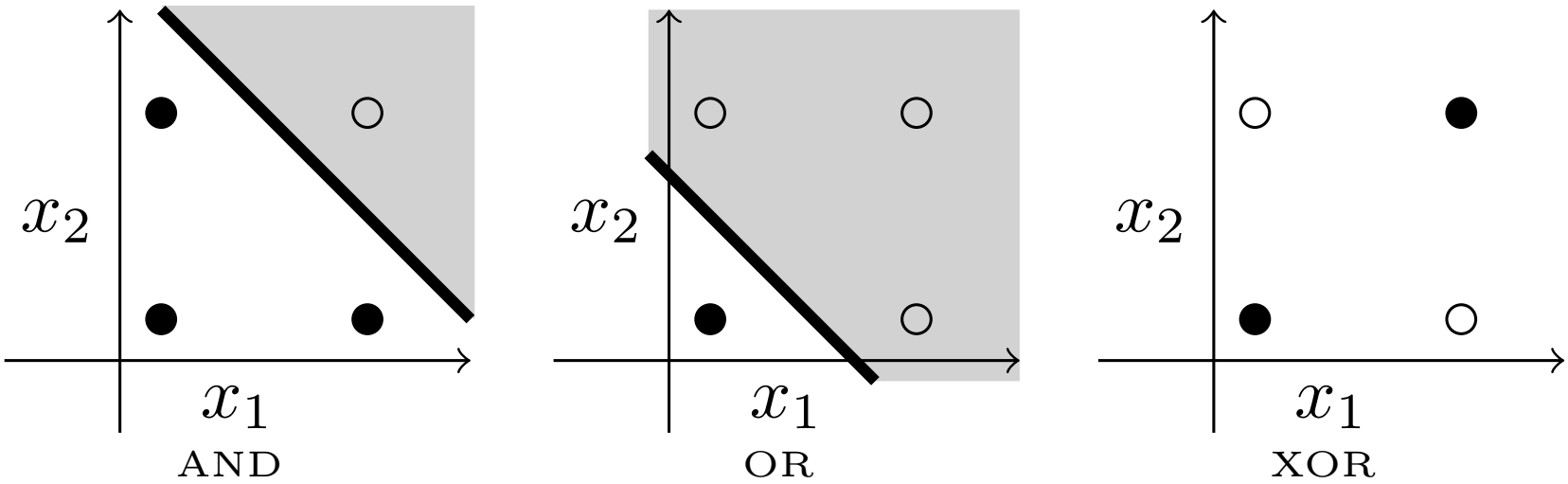


Figure 8: The logical AND and OR functions are linearly separable, but the XOR is not. This figure is Figure 4.2 of (Kelleher, 2019) and is used here with permission.

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } z \geq 1 \\ 0 & otherwise \end{cases} \tag{12}$$

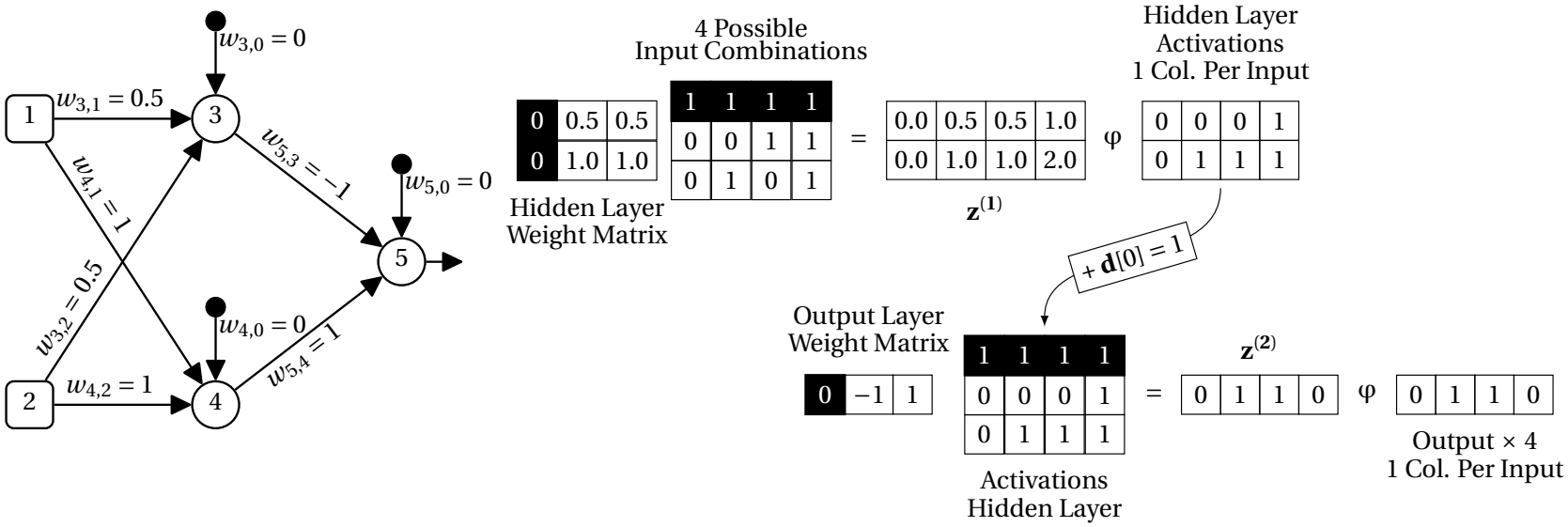


Figure 9: (left) The XOR function implemented as a two-layer neural network. (right) The network processing the four possible input combinations, one combination plus bias input per column: [*bias*, *FALSE*,*FALSE*] $\rightarrow$ [1,0,0]; [*bias*, *FALSE*,*TRUE*] $\rightarrow$ [1,0,1]; [*bias*, *TRUE*,*FALSE*] $\rightarrow$ [1,1,0]; [*bias*, *TRUE*,*TRUE*] $\rightarrow$ [1,1,1].

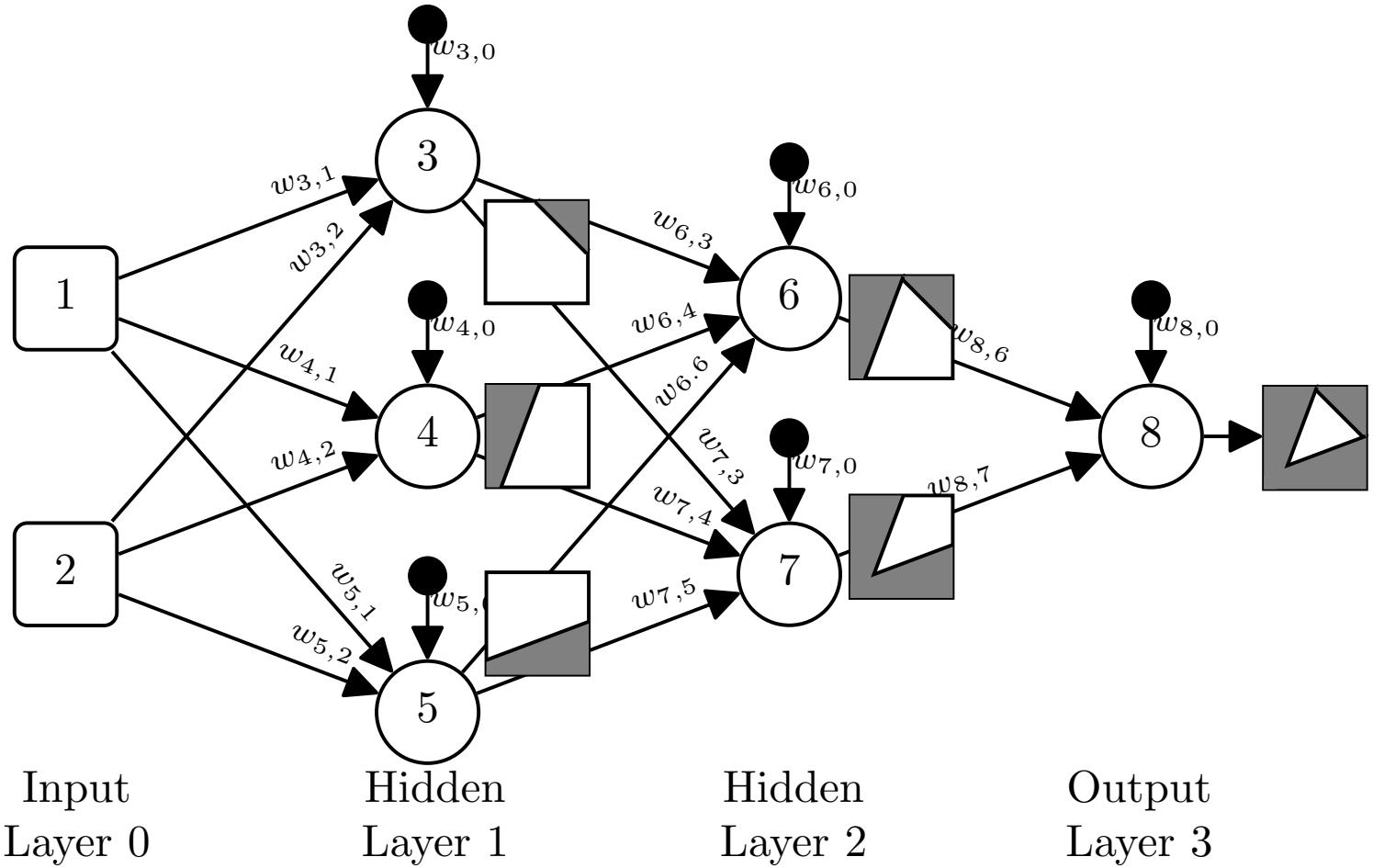


Figure 10: An illustration of how the representational capacity of a network increases as more layers are added to the network. This figure was inspired by Figure 4.2 in (Reed and Marks, 1999) and Figure 3.10 in (Marsland, 2011).

Standard Approach: Backpropagation and Gradient Descent

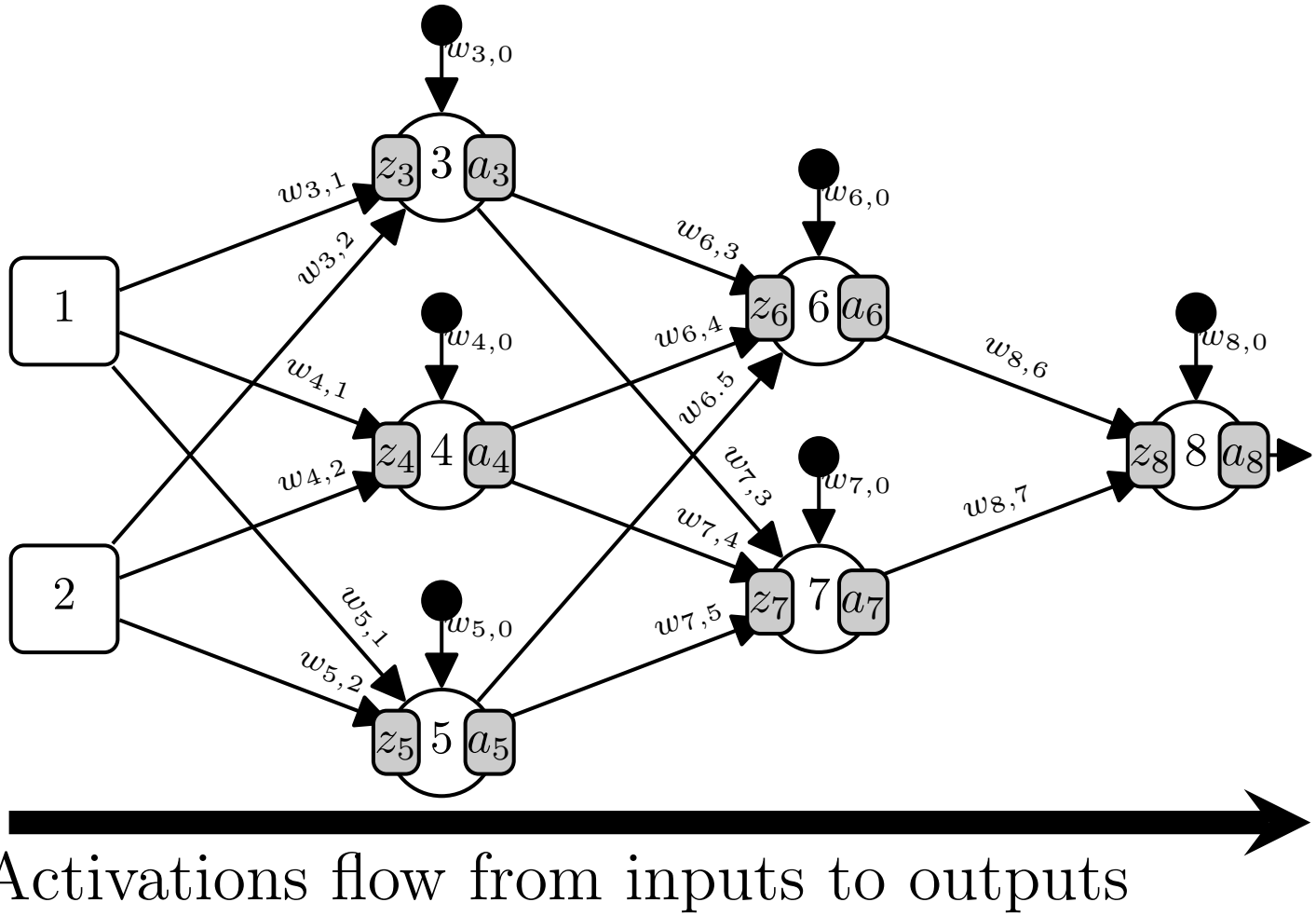


Figure 11: The calculation of the z values and activations of each neuron during the forward pass of the backpropagation algorithm. This figure is based on Figure 6.5 of (Kelleher, 2019).

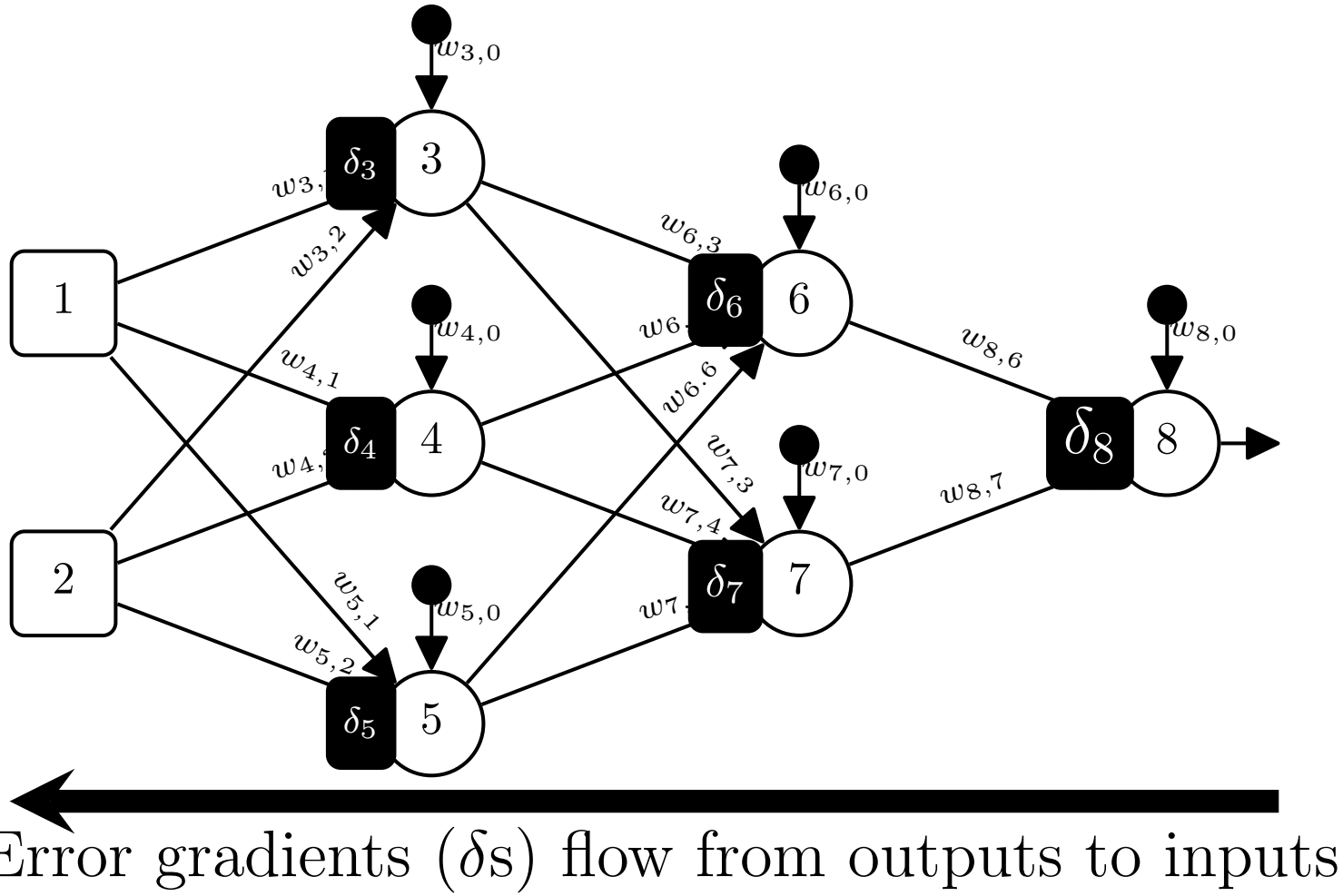


Figure 12: The backpropagation of the δ values during the backward pass of the backpropagation algorithm. This figure is based on Figure 6.6 of (Kelleher, 2019).

$$\boldsymbol{\delta}_k = \frac{\partial \mathcal{E}}{\partial z_k} \tag{13}$$

$$\boldsymbol{\delta}_k = \frac{\partial a_k}{\partial z_k} \times \frac{\partial \mathcal{E}}{\partial a_k} \tag{14}$$

$$\frac{d}{dz} logistic(z) = logistic(z) \times (1 - logistic(z)) \tag{15}$$

$$\begin{aligned} \frac{d}{dz} logistic(z = 0) &= logistic(0) \times (1 - logistic(0)) \\ &= 0.5 \times (1 - 0.5) \\ &= 0.25 \end{aligned} \tag{16}$$

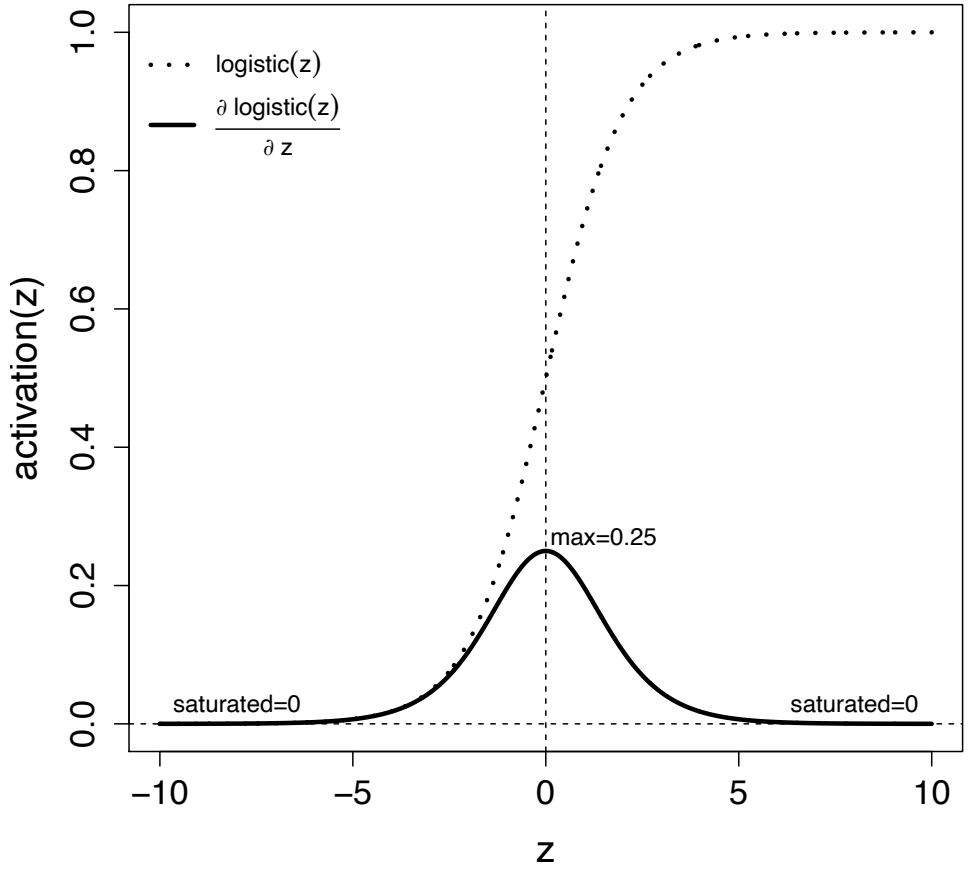


Figure 13: Plots of the logistic function and its derivative. This figure is Figure 4.6 of (Kelleher, 2019) and is used here with permission.

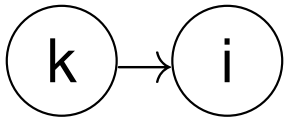
$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2} \sum_{i=1}^{n} \left(t_i - \mathbb{M}_{\mathbf{w}} \left(\mathbf{d}_i\right)\right)^2 \tag{17}$$

$$\frac{\partial}{\partial \mathbf{w}\left[j\right]} L_2\left(\mathbb{M}_{\mathbf{w}}, \mathbf{d}\right) = \left(t - \mathbb{M}_{\mathbf{w}}(\mathbf{d})\right) \times -\mathbf{d}\left[j\right] \tag{18}$$

$$\frac{\partial \mathcal{E}}{\partial a_k} = \frac{\partial L_2\left(\mathbb{M}_{\mathbf{w}}, \mathbf{d}\right)}{\partial \mathbb{M}_{\mathbf{w}}(\mathbf{d}))} = t - \mathbb{M}_{\mathbf{w}}(\mathbf{d}) = t_k - a_k \tag{19}$$

$$\frac{\partial \mathcal{E}}{\partial a_k} = -\left(t_k - a_k\right) \tag{20}$$

$$
\begin{aligned}
\boldsymbol{\delta}_k &= \frac{\partial a_k}{\partial z_k} \times \frac{\partial \mathcal{E}}{\partial a_k} \\
&= \frac{\partial a_k}{\partial z_k} \times -(t_k - a_k) \\
&= \underbrace{\frac{d}{dz} logistic(z)}_{\text{Assuming a logistic activation function}} \times -(t_k - a_k) \\
&= \underbrace{(logistic(z) \times (1 - logistic(z)))}_{\text{Assuming a logistic activation function}} \times -(t_k - a_k) \qquad (21)
\end{aligned}
$$



$$\frac{\partial \mathcal{E}}{\partial a_k} = \sum_{i=1}^{n} w_{i,k} \times \boldsymbol{\delta}_i \tag{22}$$

$$\begin{aligned}
\boldsymbol{\delta}_k &= \frac{\partial a_k}{\partial z_k} \times \frac{\partial \mathcal{E}}{\partial a_k} \\
&= \frac{\partial a_k}{\partial z_k} \times \left(\sum_{i=1}^{n} w_{i,k} \times \boldsymbol{\delta}_i \right) \\
&= \underbrace{\frac{d}{dz} logistic(z)}_{\text{Assuming a logistic activation function}} \times \left(\sum_{i=1}^{n} w_{i,k} \times \boldsymbol{\delta}_i \right) \\
&= \underbrace{(logistic(z) \times (1 - logistic(z)))}_{\text{Assuming a logistic activation function}} \times \left(\sum_{i=1}^{n} w_{i,k} \times \boldsymbol{\delta}_i \right) \qquad (23)
\end{aligned}$$

$$\frac{\partial \mathcal{E}}{\partial w_{i,k}} = \frac{\partial \mathcal{E}}{\partial a_i} \times \frac{\partial a_i}{\partial z_i} \times \frac{\partial z_i}{\partial w_{i,k}} \tag{24}$$

$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial w_{i,k}} &= \frac{\partial \mathcal{E}}{\partial a_i} \times \frac{\partial a_i}{\partial z_i} \times \frac{\partial z_i}{\partial w_{i,k}} \\ &= \boldsymbol{\delta}_i \times \frac{\partial z_i}{\partial w_{i,k}} \end{aligned} \tag{25}$$

$$\frac{\partial z_i}{\partial w_{i,k}} = a_k \tag{26}$$

$$\begin{aligned}\frac{\partial \mathcal{E}}{\partial w_{i,k}} &= \frac{\partial \mathcal{E}}{\partial a_i} \times \frac{\partial a_i}{\partial z_i} \times \frac{\partial z_i}{\partial w_{i,k}} \\ &= \boldsymbol{\delta}_i \times \frac{\partial z_i}{\partial w_{i,k}} \\ &= \boldsymbol{\delta}_i \times a_k \end{aligned} \tag{27}$$

$$w_{i,k} \leftarrow w_{i,k} - \alpha \times \boldsymbol{\delta}_i \times a_k \tag{28}$$

$$\Delta w_{i,k} = \sum_{j=1}^{m} \boldsymbol{\delta}_{i,j} \times a_{k,j} \tag{29}$$

$$w_{i,k} \leftarrow w_{i,k} - \alpha \times \Delta w_{i,k} \tag{30}$$

Require: set of training instances $\mathcal{D}$
Require: a learning rate α that controls how quickly the algorithm converges
Require: a batch size B specifying the number of examples in each batch
Require: a convergence criterion

Shuffle $\mathcal{D}$ and create the mini-batches: $[(\mathbf{X}^{(1)}, \mathbf{Y}^{(1)}), \ldots, (\mathbf{X}^{k}, \mathbf{Y}^{k})]$
Initialize the weight matrices for each layer: $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}$
repeat ▷ Each repeat loop is one epoch
  for t=1 to number of mini-batches **do** ▷ Each for loop is one iteration
    $\mathbf{A}^{(0)} \leftarrow \mathbf{X}^{(t)}$
    for l=1 to L **do**
      $\mathbf{v} \leftarrow [1_0, \ldots 1_m]$ ▷ Create $\mathbf{v}$ the vector of bias terms
      $\mathbf{A}^{(l-1)} \leftarrow [\mathbf{v}; \mathbf{A}^{(l-1)}]$ ▷ Insert $\mathbf{v}$ into the activation matrix
      $\mathbf{Z}^{(l)} \leftarrow \mathbf{W}^{l}\mathbf{A}^{(l-1)}$
      $\mathbf{A}^{(l)} \leftarrow \varphi(\mathbf{Z}^{(l)})$ ▷ Elementwise application of φ to $\mathbf{Z}^{(l)}$
    end for
    for each weight $w_{i,k}$ in the network **do**
      $\Delta w_{i,k} = 0$
    end for
    for each example in the mini-batch **do** ▷ Backpropagate the δs
      for each neuron i in the output layer **do**
        $\delta_i = \frac{\partial \mathcal{E}}{\partial a_i} \times \frac{\partial a_i}{\partial z_i}$ ▷ See Equation (21)[28]
      end for
      for l = L-1 to 1 **do**
        for each neuron i in the layer l **do**
          $\delta_i = \frac{\partial \mathcal{E}}{\partial a_i} \times \frac{\partial a_i}{\partial z_i}$ ▷ See Equation (23)[29]
        end for
      end for
      for each weight $w_{i,k}$ in the network **do**
        $\Delta w_{i,k} = \Delta w_{i,k} + (\delta_i \times a_k)$ ▷ Equation (29)[33]
      end for
    end for
    for each weight $w_{i,k}$ in the network **do**
      $w_{i,k} \leftarrow w_{i,k} - \alpha \times \Delta w_{i,k}$ ▷ Equation (30)[33]
    end for
  end for
  shuffle($[(\mathbf{X}^{(1)}, \mathbf{Y}^{(1)}), \ldots, (\mathbf{X}^{k}, \mathbf{Y}^{k})]$)
until convergence occurs

Table 1: Hourly samples of ambient factors and full load electrical power output of a combined cycle power plant.

ID	Ambient Temperature °C	Relative Humidity %	Electrical Output MW
1	03.21	86.34	491.35
2	31.41	68.50	430.37
3	19.31	30.59	463.00
4	20.64	99.97	447.14

Table 2: The minimum and maximum values for the AMBIENT TEMPERATURE, RELATIVE HUMIDITY, and ELECTRICAL OUTPUT features in the power plant dataset.

	AMBIENT TEMPERATURE	RELATIVE HUMIDITY	ELECTRICAL OUTPUT
Min	$1.81°$C	25.56%	420.26MW
Max	$37.11°$C	100.16%	495.76MW

Table 3: The *range-normalized* hourly samples of ambient factors and full load electrical power output of a combined cycle power plant, rounded to two decimal places.

ID	Ambient Temperature °C	Relative Humidity %	Electrical Output MW
1	0.04	0.81	0.94
2	0.84	0.58	0.13
3	0.50	0.07	0.57
4	0.53	1.00	0.36

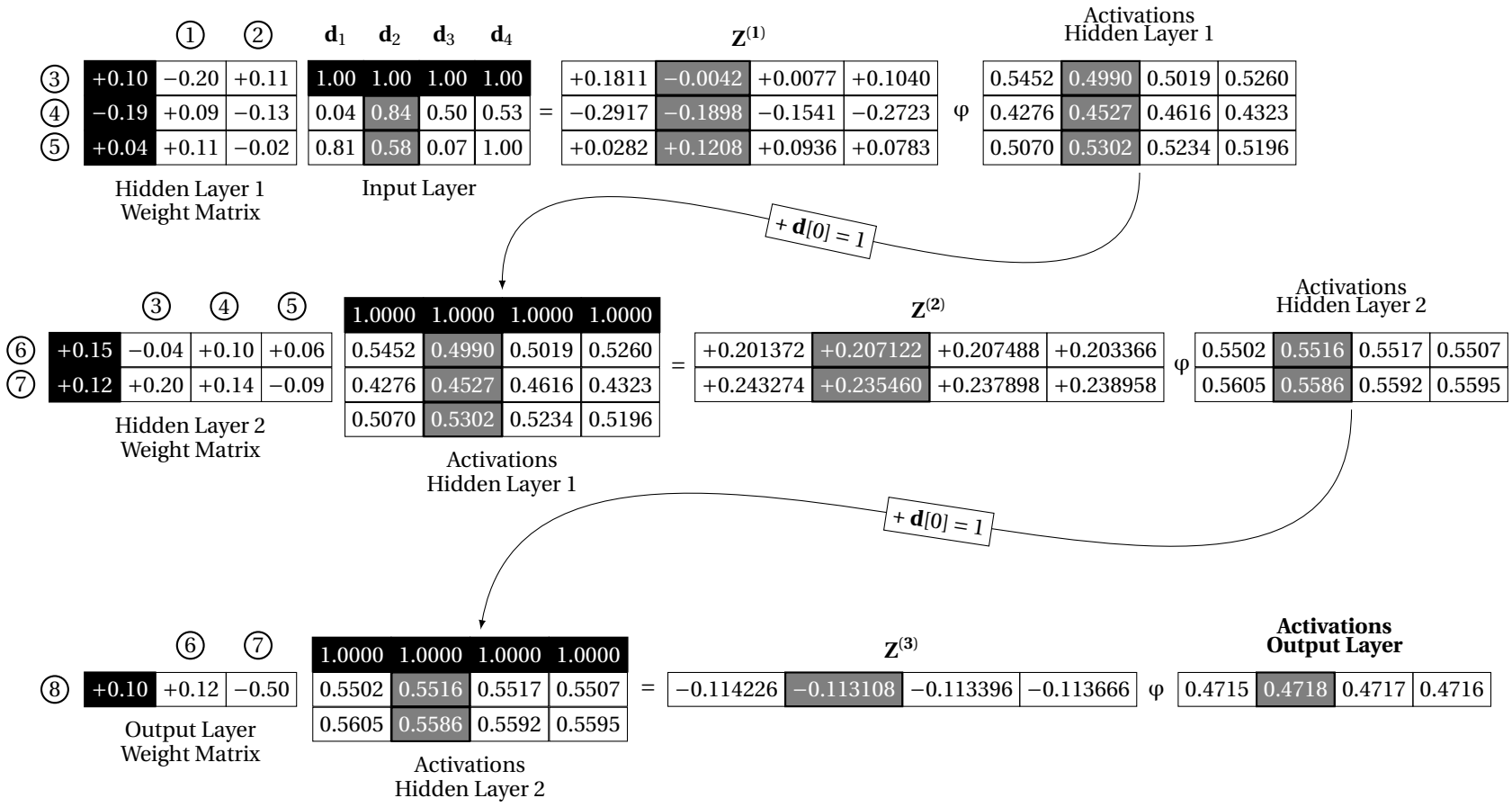


Figure 14: The forward pass of the examples listed in Table 3[38] through the network in Figure 4[11].

Table 4: The per example error after the forward pass illustrated in Figure 14[39], the per example $\partial \mathcal{E} / \partial a_8$, and the **sum of squared errors** for the model over the dataset of four examples.

	$\mathbf{d}_1$	$\mathbf{d}_2$	$\mathbf{d}_3$	$\mathbf{d}_4$
Target	0.9400	0.1300	0.5700	0.3600
Prediction	0.4715	0.4718	0.4717	0.4716
Error	0.4685	-0.3418	0.0983	-0.1116
$\partial \mathcal{E} / \partial a_8$: Error $\times -1$	-0.4685	0.3418	-0.0983	0.1116
Error^2	0.21949225	0.11682724	0.00966289	0.01245456
SSE:				0.17921847

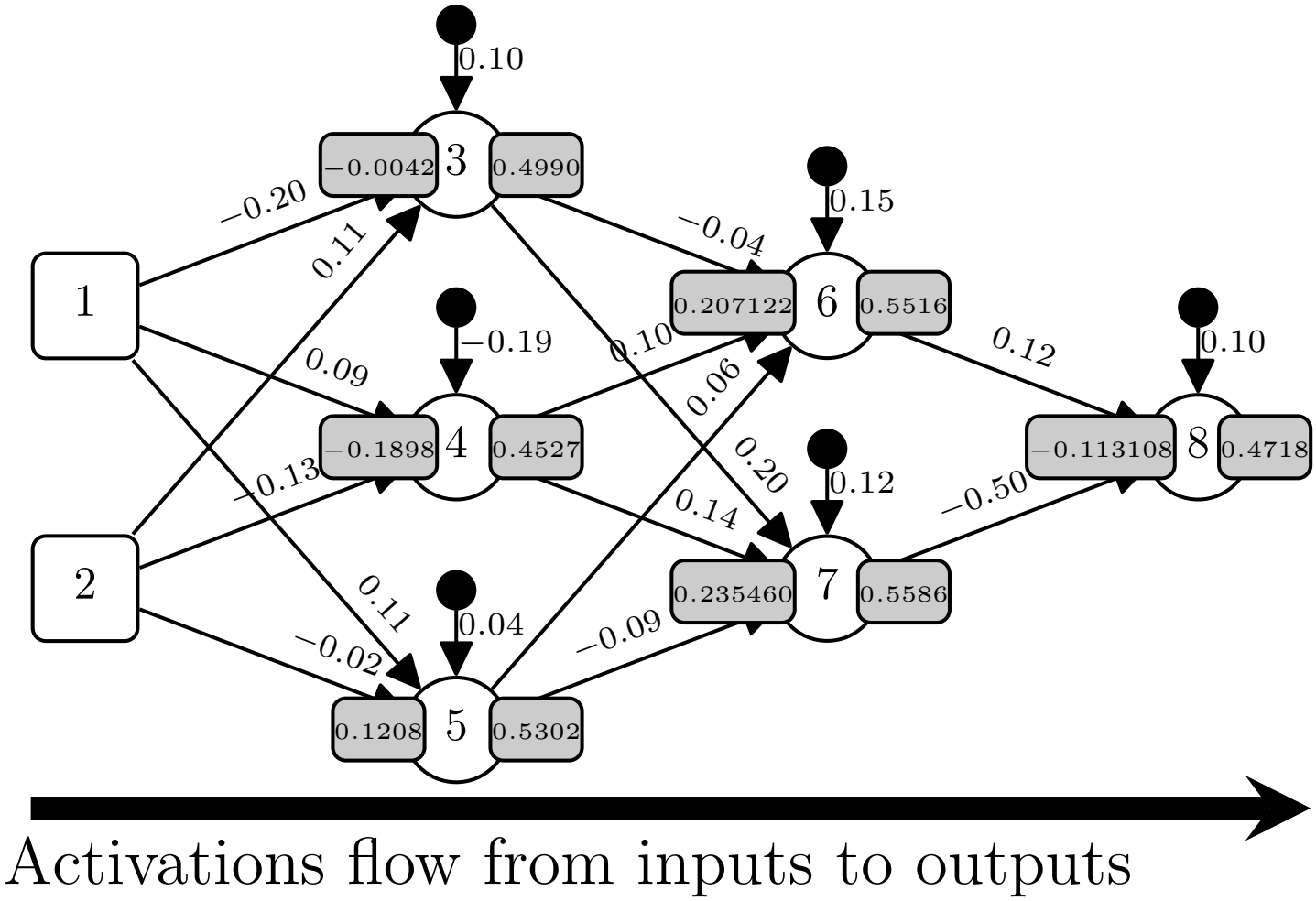


Figure 15: An illustration of the forward propagation of $\mathbf{d}_2$ through the network showing the weights on each connection, and the weighted sum z and activation a value for each neuron in the network.

$$\frac{\partial \mathcal{E}}{\partial a} = 0.3418 \tag{31}$$

$$\begin{aligned} \delta_8 = & \frac{\partial \mathcal{E}}{\partial a_8} \times \frac{\partial a_8}{\partial z_8} \\ =& 0.3418 \times 0.2492 \\ =& 0.0852 \end{aligned} \tag{32}$$

Table 5: The $\partial a/\partial z$ for each neuron for Example 2 rounded to four decimal places.

NEURON	z	$\partial a/\partial z$
3	-0.004200	0.2500
4	-0.189800	0.2478
5	0.120800	0.2491
6	0.207122	0.2473
7	0.235460	0.2466
8	-0.113108	0.2492

$$
\begin{aligned}
\delta_6 = & \frac{\partial \mathcal{E}}{\partial a_6} \times \frac{\partial a_6}{\partial z_6} \\
= & \left(\sum \delta_i \times w_{i,6} \right) \times \frac{\partial a_6}{\partial z_6} \\
= & \left(\delta_8 \times w_{8,6} \right) \times \frac{\partial a_6}{\partial z_6} \\
= & (0.0852 \times 0.12) \times 0.2473 \\
= & \quad 0.0025
\end{aligned} \tag{33}
$$

$$
\begin{aligned}
\delta_7 = & \frac{\partial \mathcal{E}}{\partial a_7} \times \frac{\partial a_7}{\partial z_7} \\
= & \left(\sum \delta_i \times w_{i,7} \right) \times \frac{\partial a_7}{\partial z_7} \\
= & \left(\delta_8 \times w_{8,7} \right) \times \frac{\partial a_6}{\partial z_6} \\
= & (0.0852 \times -0.50) \times 0.2466 \\
= & -0.0105
\end{aligned} \tag{34}
$$

$$
\begin{aligned}
\delta_3 = & \frac{\partial \mathcal{E}}{\partial a_3} \times \frac{\partial a_3}{\partial z_3} \\
= & \left(\sum \delta_i \times w_{i,3} \right) \times \frac{\partial a_3}{\partial z_3} \\
= & \left((\delta_6 \times w_{6,3}) + (\delta_7 \times w_{7,3}) \right) \times \frac{\partial a_3}{\partial z_3} \\
= & ((0.0025 \times -0.04) + (-0.0105 \times 0.20)) \times 0.2500 \\
= & -0.0006
\end{aligned} \tag{35}
$$

$$
\begin{aligned}
\delta_4 = & \frac{\partial \mathcal{E}}{\partial a_4} \times \frac{\partial a_4}{\partial z_4} \\
= & \left(\sum \delta_i \times w_{i,4} \right) \times \frac{\partial a_4}{\partial z_4} \\
= & \left((\delta_6 \times w_{6,4}) + (\delta_7 \times w_{7,4}) \right) \times \frac{\partial a_4}{\partial z_4} \\
= & ((0.0025 \times 0.10) + (-0.0105 \times 0.14)) \times 0.2478 \\
= & -0.0003
\end{aligned} \tag{36}
$$

$$
\begin{aligned}
\delta_5 = & \frac{\partial \mathcal{E}}{\partial a_5} \times \frac{\partial a_5}{\partial z_5} \\
= & \left(\sum \delta_i \times w_{i,5} \right) \times \frac{\partial a_5}{\partial z_5} \\
= & \left((\delta_6 \times w_{6,5}) + (\delta_7 \times w_{7,5}) \right) \times \frac{\partial a_5}{\partial z_5} \\
= & ((0.0025 \times 0.06) + (-0.0105 \times -0.09)) \times 0.2491 \\
= & \quad 0.0003 \qquad (37)
\end{aligned}
$$

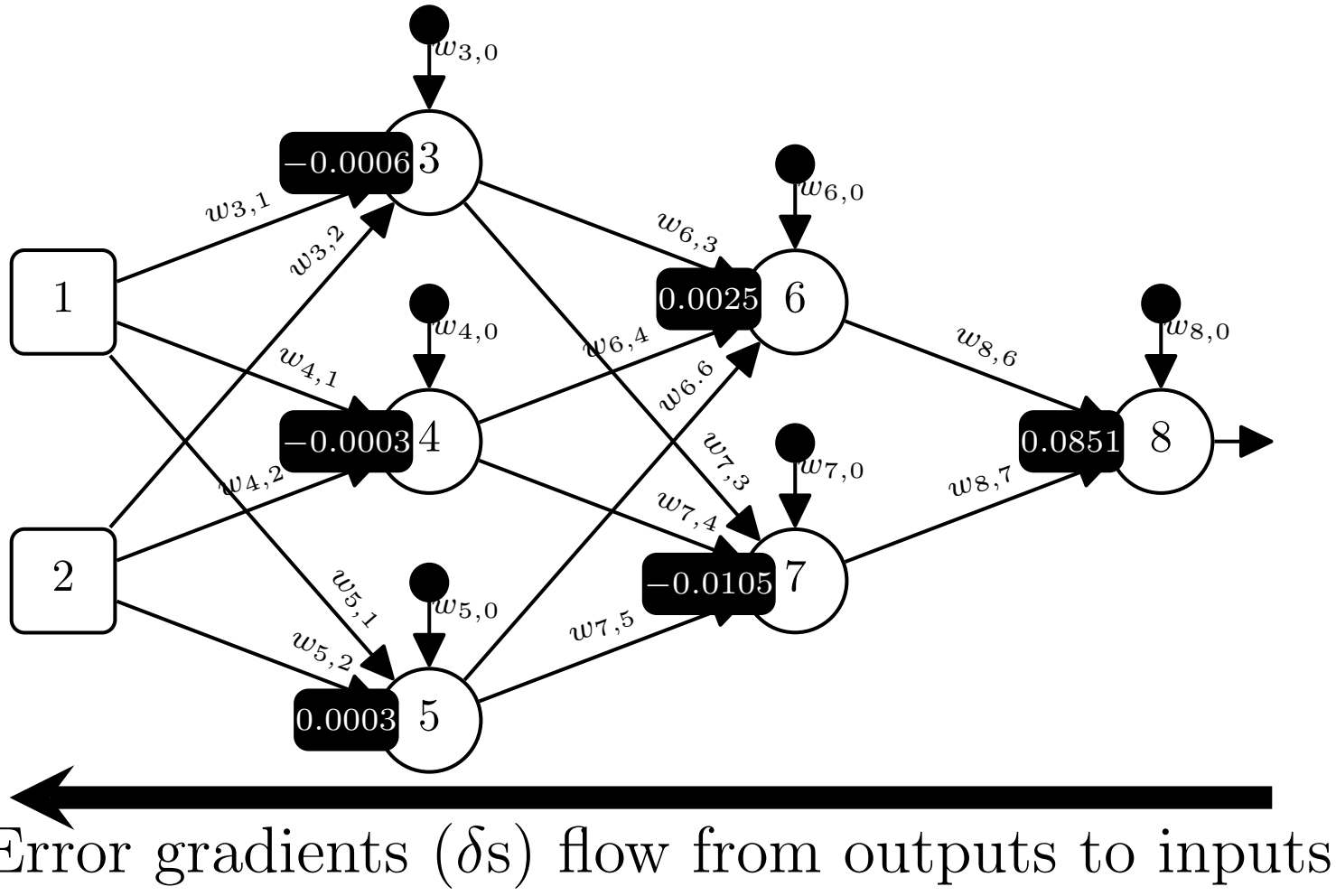


Figure 16: The δs for each of the neurons in the network for Ex. 2.

Table 6: The $\partial \mathcal{E}/\partial w_{i,k}$ calculations for $\mathbf{d}_2$ for every weight in the network. The neuron index 0 denotes the bias input for each neuron.

NEURON$_i$	NEURON$_k$	$w_{i,k}$	δ_i	a_k	$\partial \mathcal{E}/\partial w_{i,k}$
8	0	$w_{8,0}$	0.0852	1	$0.0852 \times 1 = 0.0852$
8	6	$w_{8,6}$	0.0852	0.5516	$0.0852 \times 0.5516 = 0.04699632$
8	7	$w_{8,7}$	0.0852	0.5586	$0.0852 \times 0.5586 = 0.04759272$
7	0	$w_{7,0}$	-0.0105	1	$-0.0105 \times 1 = -0.0105$
7	3	$w_{7,3}$	-0.0105	0.4990	$-0.0105 \times 0.4527 = -0.0052395$
7	4	$w_{7,4}$	-0.0105	0.4527	$-0.0105 \times 0.4527 = -0.00475335$
7	5	$w_{7,5}$	-0.0105	0.5302	$-0.0105 \times 0.5302 = -0.0055671$
6	0	$w_{6,0}$	0.0025	1	$0.0025 \times 1 = 0.0025$
6	3	$w_{6,3}$	0.0025	0.4990	$0.0025 \times 0.4527 = 0.0012475$
6	4	$w_{6,4}$	0.0025	0.4527	$0.0025 \times 0.4527 = 0.00113175$
6	5	$w_{6,5}$	0.0025	0.5302	$0.0025 \times 0.5302 = 0.0013255$
5	0	$w_{5,0}$	0.0003	1	$0.0003 \times 1 = 0.0003$
5	1	$w_{5,1}$	0.0003	0.84	$0.0003 \times 0.84 = 0.000252$
5	2	$w_{5,2}$	0.0003	0.58	$0.0003 \times 0.58 = 0.000174$
4	0	$w_{4,0}$	-0.0003	1	$-0.0003 \times 1 = -0.0003$
4	1	$w_{4,1}$	-0.0003	0.84	$-0.0003 \times 0.84 = -0.000252$
4	2	$w_{4,2}$	-0.0003	0.58	$-0.0003 \times 0.58 = -0.000174$
3	0	$w_{3,0}$	-0.0006	1	$-0.0006 \times 1 = -0.0006$
3	1	$w_{3,1}$	-0.0006	0.84	$-0.0006 \times 0.84 = -0.000504$
3	2	$w_{3,2}$	-0.0006	0.58	$-0.0006 \times 0.58 = -0.000348$

$$
\begin{aligned}
w_{7,5} &= w_{7,5} - \alpha \times \boldsymbol{\delta}_7 \times a_5 \\
&= w_{7,5} - \alpha \times \frac{\partial \mathcal{E}}{\partial w_{i,k}} \\
&= -0.09 - 0.2 \times -0.0055671 \\
&= -0.08888658
\end{aligned} \tag{38}
$$

Table 7: The calculation of $\Delta w_{7,5}$ across our four examples.

MINI-BATCH EXAMPLE	$\frac{\partial \mathcal{E}}{\partial w_{7,5}}$
$\mathbf{d}_1$	0.00730080
$\mathbf{d}_2$	-0.00556710
$\mathbf{d}_3$	0.00157020
$\mathbf{d}_4$	-0.00176664
$\Delta w_{7,5}$=	0.00153726

$$\begin{aligned} w_{7,5} &= w_{7,5} - \alpha \times \Delta w_{i,k} \\ &= -0.09 - 0.2 \times 0.00153726 \\ &= -0.0903074520 \end{aligned} \tag{39}$$

Table 8: The per example error after each weight has been updated once, the per example $\partial \mathcal{E}/\partial a_8$, and the **sum of squared errors** for the model.

	$\mathbf{d}_1$	$\mathbf{d}_2$	$\mathbf{d}_3$	$\mathbf{d}_4$
Target	0.9400	0.1300	0.5700	0.3600
Prediction	0.4738	0.4741	0.4740	0.4739
Error	0.4662	-0.3441	0.0960	-0.1139
$\partial \mathcal{E}/\partial a_8$: Error $\times -1$	-0.4662	0.3441	-0.0960	0.1139
Error^2	0.21734244	0.11840481	0.009216	0.01297321
SSE:				0.17896823

Table 9: The per example prediction, error, and the sum of squared errors after training has converged to an $SSE < 0.0001$.

	$\mathbf{d}_1$	$\mathbf{d}_2$	$\mathbf{d}_2$	$\mathbf{d}_2$
Target	0.9400	0.1300	0.5700	0.3600
Prediction	0.9266	0.1342	0.5700	0.3608
Error	0.0134	-0.0042	0.0000	-0.0008
Error^2	0.00017956	0.00001764	0.00000000	0.00000064
SSE:				0.00009892

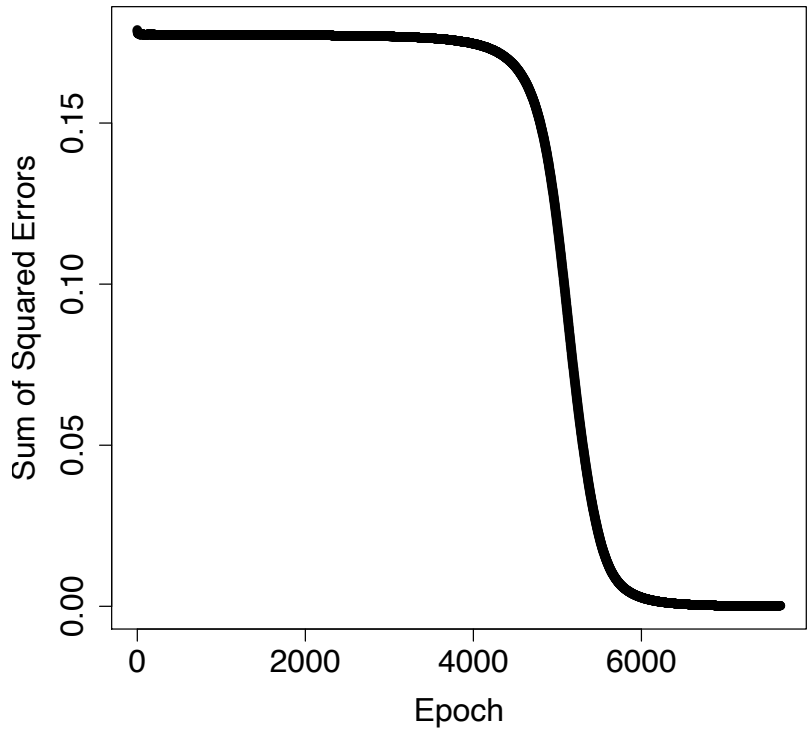


Figure 17: A plot showing how the sum of squared errors of the network changed during training.