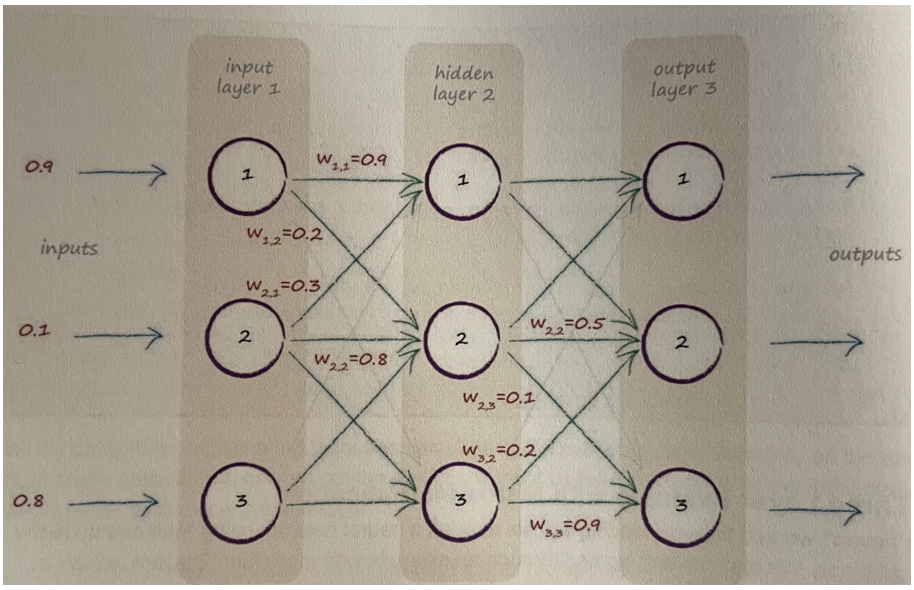
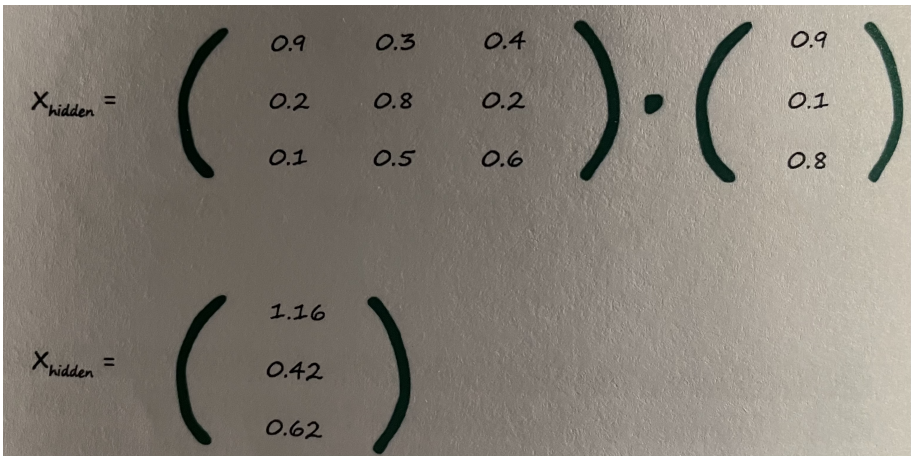
Make Your Own NN p64-p79

2021.1.30



The first layer is the input layer. The final layer is output layer. The middle layer is called the hidden layer. Recall that $\mathbf{X} = \mathbf{W} \cdot \mathbf{I}$, where $\mathbf{I}$ is the input matrix.

$$I = \begin{pmatrix} 0.9 \\ 0.1 \\ 0.8 \end{pmatrix}, \; X_{\text{hidden}} = W_{\text{input\_hidden}} \cdot I$$



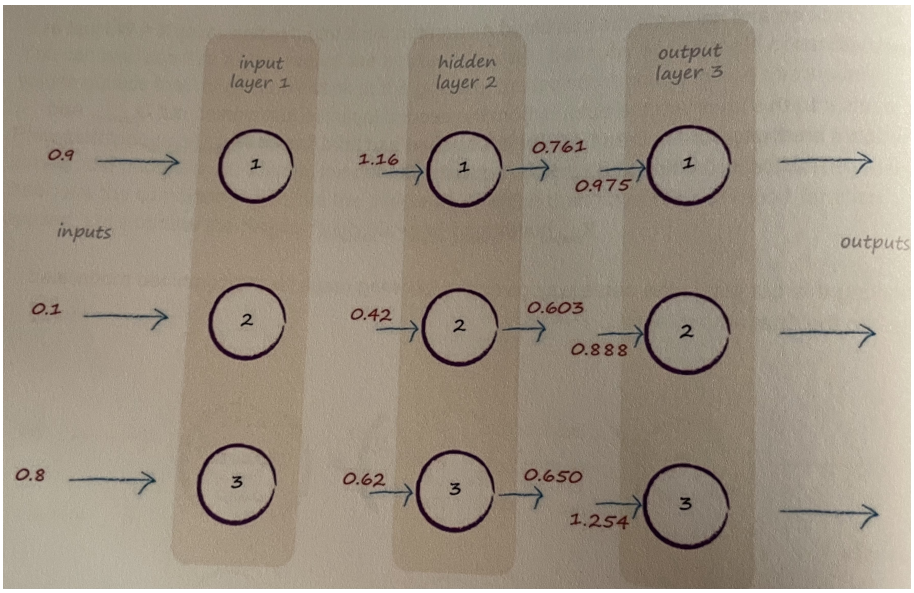
Remember those nodes apply a **sigmoid** activation function.

$$\mathbf{O}_{hidden} = sigmoid\ (\ \mathbf{X}_{hidden}\ )$$

$$\mathbf{O}_{hidden} = sigmoid \begin{pmatrix} 1.16 \\ 0.42 \\ 0.62 \end{pmatrix}$$

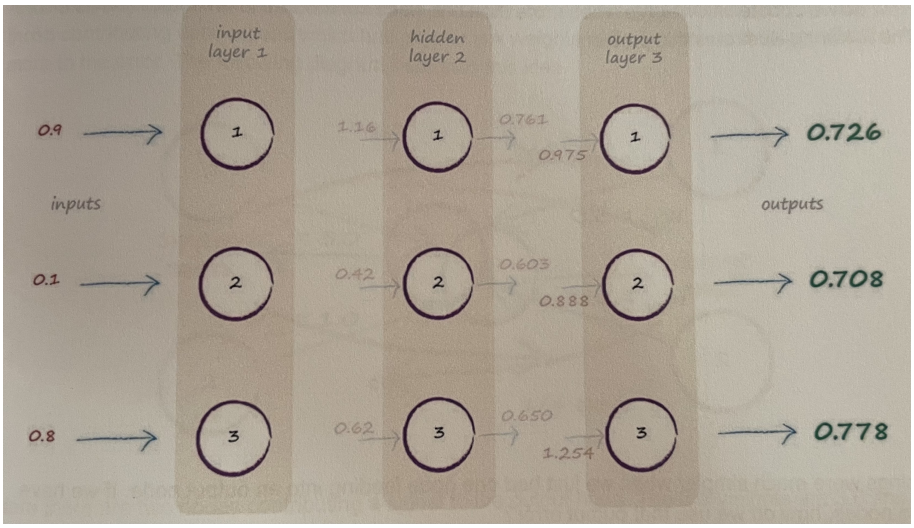
$$\mathbf{O}_{hidden} = \begin{pmatrix} 0.761 \\ 0.603 \\ 0.650 \end{pmatrix}$$

$$X_{\text{output}} = W_{\text{hidden\_output}} \cdot O_{\text{hidden}}$$

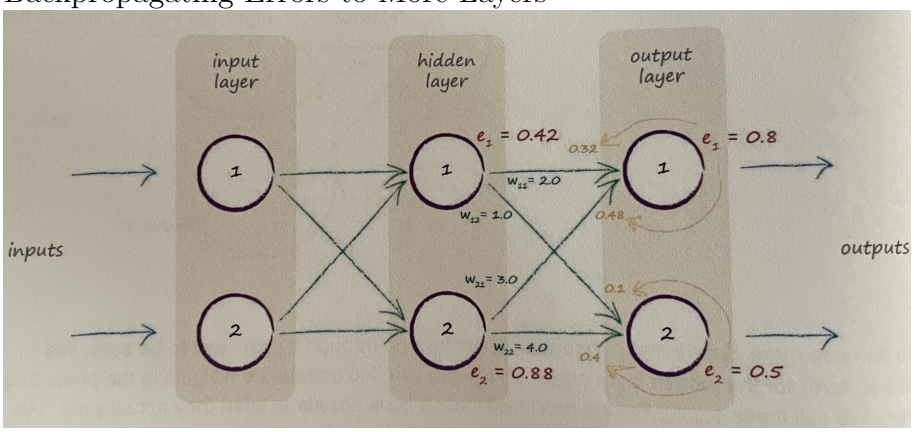


$$\mathbf{O}_{output} = sigmoid \begin{pmatrix} 0.975 \\ 0.888 \\ 1.254 \end{pmatrix}$$

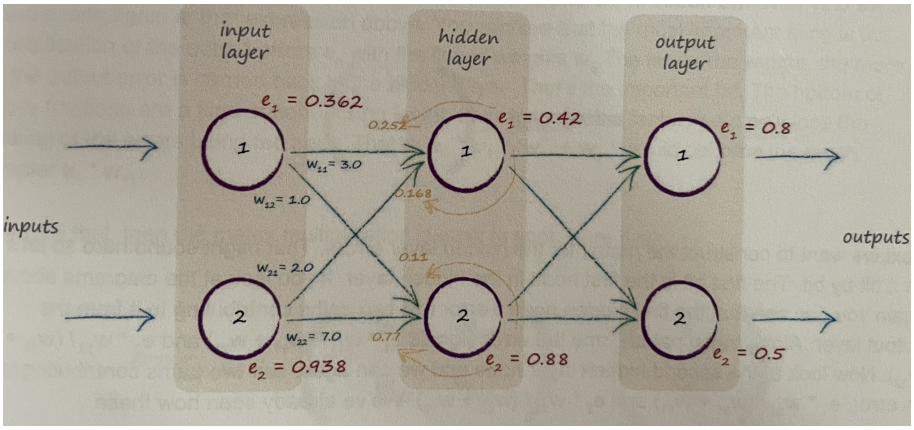
$$\mathbf{O}_{output} = \begin{pmatrix} 0.726 \\ 0.708 \\ 0.778 \end{pmatrix}$$



Backpropagating Errors to More Layers



$$
\begin{aligned}
e_{hidden,1} &= \text{sum of split errors on links } w_{11} \text{ and } w_{22} \\
&= e_{output,1} * \frac{w_{11}}{w_{11} + w_{21}} + e_{output,2} * \frac{w_{12}}{w_{12} + w_{22}}
\end{aligned}
$$



Key Points

⋆ NN learns by refining their link wights. This is guided by the error - the difference betwwen the tight answers given by the training data and their actual output

⋆ However, the error associated with internal nodes is not obvious. One approach is to split the output layer errors in proportion to the size of the connected link weights, and then recombine these bits at each internal node