Network with 1 Hidden Layer

Input layer: \( x_1, \ldots, x_d \); \( x_{d+1} = 1 \)

Hidden units: \( h_1, \ldots, h_m \); \( h_{m+1} = 1 \)

Output layer: \( z_1, \ldots, z_k \) [We might have more than one output so we can build multiple classifiers that share hidden units.]

Layer 1 weights: \( m \times (d+1) \) matrix \( V \) \( V_i \) is row \( i \)
Layer 2 weights: \( k \times (m+1) \) matrix \( W \) \( W_i \) is row \( i \)

Recall \( s(\gamma) = \frac{1}{1 + e^{-\gamma}} \)

other nonlinear fn's can be used

For vector \( v \), \( s(v) = \begin{bmatrix} s(v_1) \\ s(v_2) \\ \vdots \end{bmatrix} \) [We apply \( s \) to a vector component by component]

\[
\begin{align*}
h_i &= s \left( \sum_{j=1}^{3} V_{ij} x_j \right) & \text{In short, } h = s(Vx) \\
z &= s(Wh) = s(Ws(Vx)) & \text{add a } 1 \text{ to end of vector}
\end{align*}
\]
Computing Gradients for Arithmetic Expressions

\[
\frac{\partial f}{\partial a} = \frac{\partial f}{\partial d} \frac{\partial d}{\partial a}
\]

\[
= \frac{\partial f}{\partial d}
\]

\[
\frac{\partial f}{\partial b} = \frac{\partial f}{\partial d} \frac{\partial d}{\partial b}
\]

\[
= \frac{\partial f}{\partial d}
\]

\[
\frac{\partial f}{\partial c} = \frac{\partial f}{\partial e} \frac{\partial e}{\partial c}
\]

\[
= d \frac{\partial f}{\partial e}
\]

d = a + b

\[
\frac{\partial d}{\partial a} = 1 \quad \frac{\partial d}{\partial b} = 1
\]

\[
e = cd
\]

\[
\frac{\partial e}{\partial c} = d \quad \frac{\partial e}{\partial d} = c
\]

\[
f = e^2
\]

\[
\frac{\partial f}{\partial e} = 2e
\]

Each value \( z \) gives a partial derivative of the form

\[
\frac{\partial f}{\partial z} = \left( \frac{\partial f}{\partial n} \frac{\partial n}{\partial z} \right)
\]

computed during forward pass

\[
\text{computed during backward pass after forward pass}
\]

"backpropagation"
[what if a unit's output goes to more than one unit?]

\[
\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial x_1} \\
= V_{11} \frac{\partial L}{\partial z_1} + V_{12} \frac{\partial L}{\partial z_2}
\]

\[
\frac{\partial L}{\partial x_2} = \frac{\partial L}{\partial z_1} \frac{\partial z_1}{\partial x_2} + \frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial x_2} \\
= V_{12} \frac{\partial L}{\partial z_1} + V_{22} \frac{\partial L}{\partial z_2}
\]

\[
\frac{\partial L}{\partial z_1} = 2(z_1 - y_1)
\]

\[
\frac{\partial L}{\partial z_2} = 2(z_2 - y_2)
\]
The Backpropagation Algorithm

Recall $s'(y) = s(y)(1-s(y))$

$h_i = s(V_i \cdot x)$, so $\nabla_{V_i} h_i = s'(V_i \cdot x) \times h_i (1-h_i) \times$ $\nabla_{W_j} z_j = s'(W_j \cdot h) \times h_j (1-z_j) \times$ $z_j (1-z_j) h$

Compute $\nabla_V L, \nabla_W L$ one row at a time.