CS 453X: Class 21

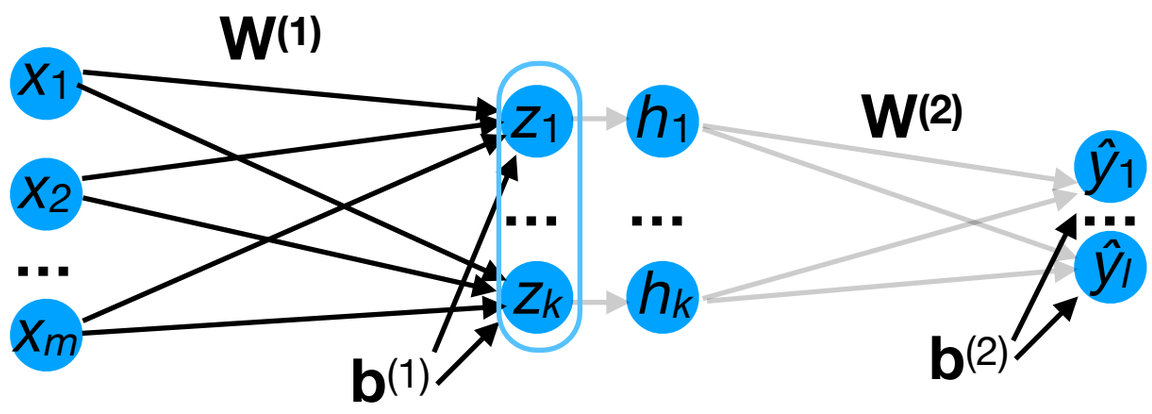
Jacob Whitehill

More on forwards and backwards propagation

Computing the gradients

- Jacobian matrices and the chain rule provide a recipe for how to compute all the gradient terms efficiently.

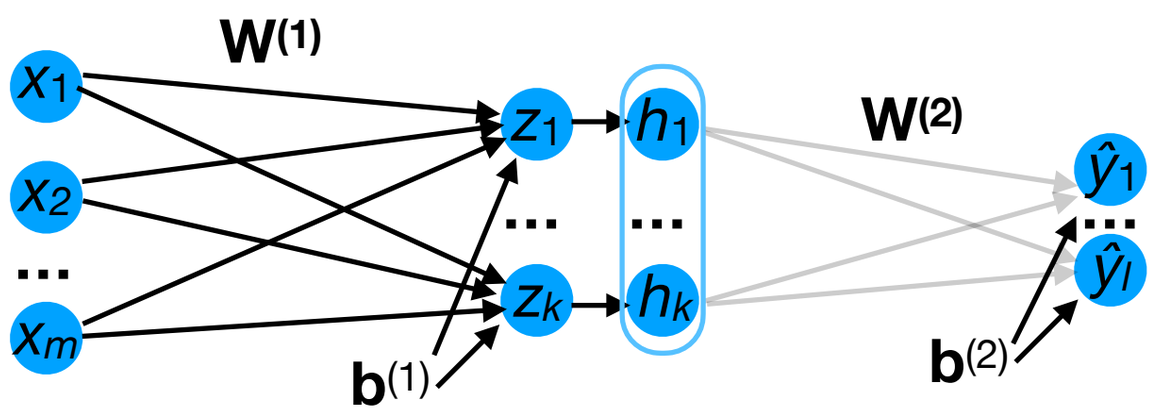
- Consider the 3-layer NN below:
  - From **x**, $\mathbf{W}^{(1)}$, and $\mathbf{b}^{(1)}$, we can compute **z**.



Computing the gradients

- Jacobian matrices and the chain rule provide a recipe for how to compute all the gradient terms efficiently.

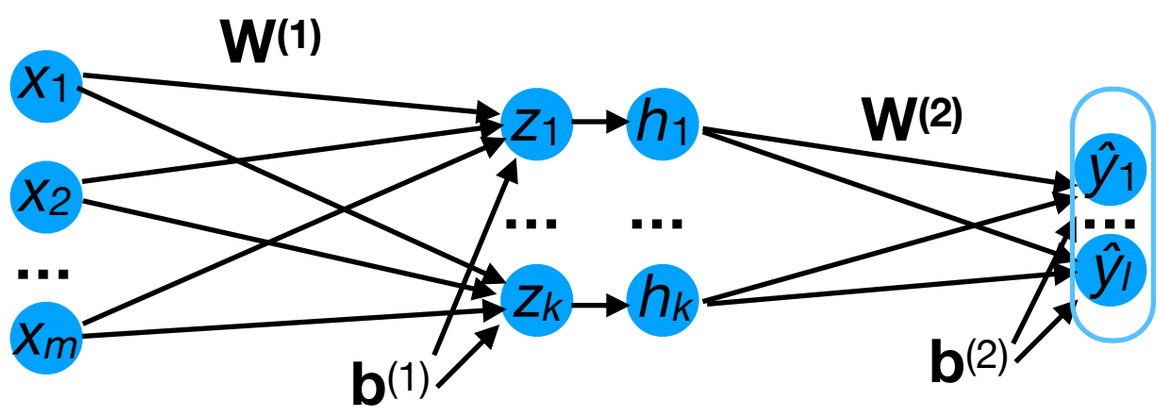
- Consider the 3-layer NN below:
  - From $\mathbf{x}$, $\mathbf{W}^{(1)}$, and $\mathbf{b}^{(1)}$, we can compute $\mathbf{z}$.
  - From $\mathbf{z}$ and σ, we can compute $\mathbf{h} = \sigma(\mathbf{z})$.



Computing the gradients

- Jacobian matrices and the chain rule provide a recipe for how to compute all the gradient terms efficiently.

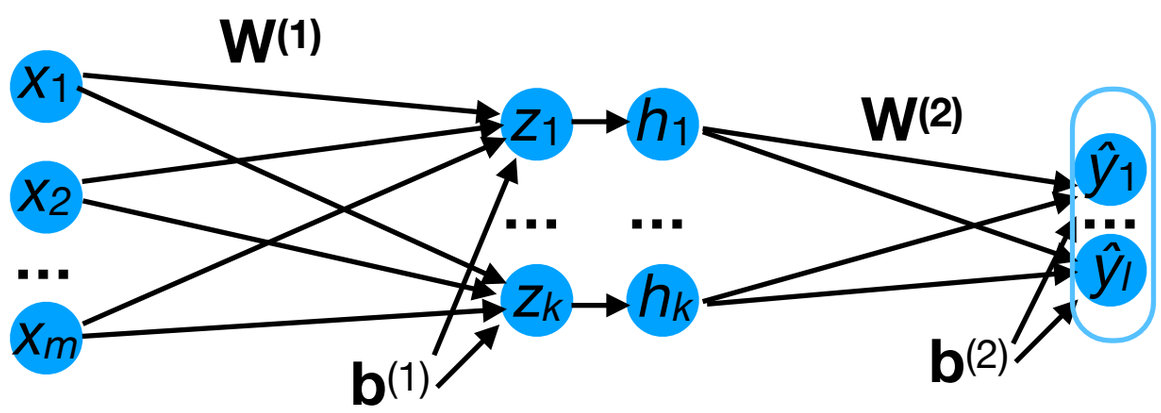
- Consider the 3-layer NN below:
  - From $\mathbf{x}$, $\mathbf{W}^{(1)}$, and $\mathbf{b}^{(1)}$, we can compute $\mathbf{z}$.
  - From $\mathbf{z}$ and σ, we can compute $\mathbf{h} = \sigma(\mathbf{z})$.
  - From $\mathbf{h}$, $\mathbf{W}^{(2)}$, and $\mathbf{b}^{(2)}$, we can compute $\hat{\mathbf{y}}$.



Computing the gradients

- Jacobian matrices and the chain rule provide a recipe for how to compute all the gradient terms efficiently.

- This process is known as **forward propagation**.
  - It produces all the intermediary (**h**, **z**) and final ($\hat{\mathbf{y}}$) network outputs.



Computing the gradients

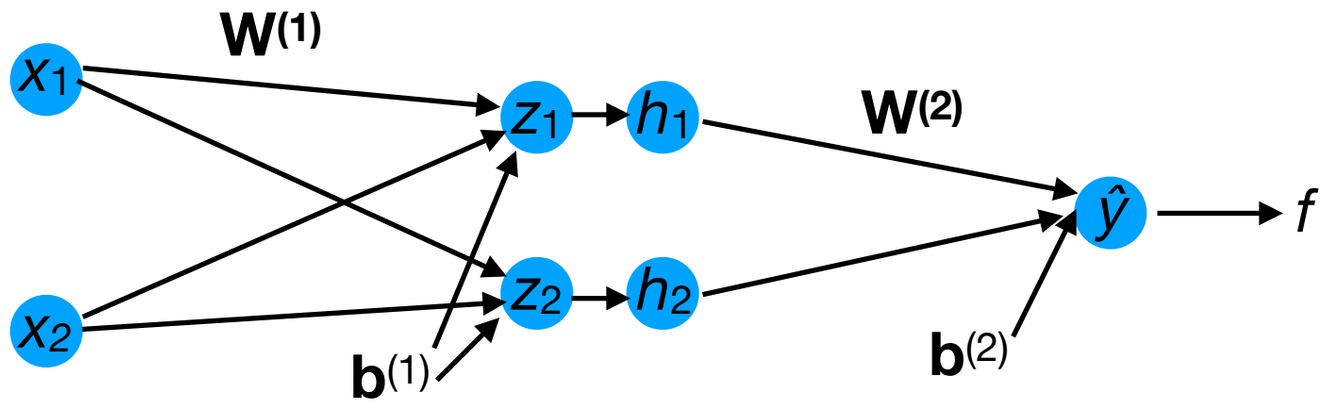
- Now, let's look at how to compute each gradient term:

$$\frac{\partial f}{\partial \mathbf{W}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{b}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{b}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

$$\frac{\partial f}{\partial \mathbf{b}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{b}^{(1)}}$$



Computing the gradients

- Now, let's look at how to compute each gradient term:

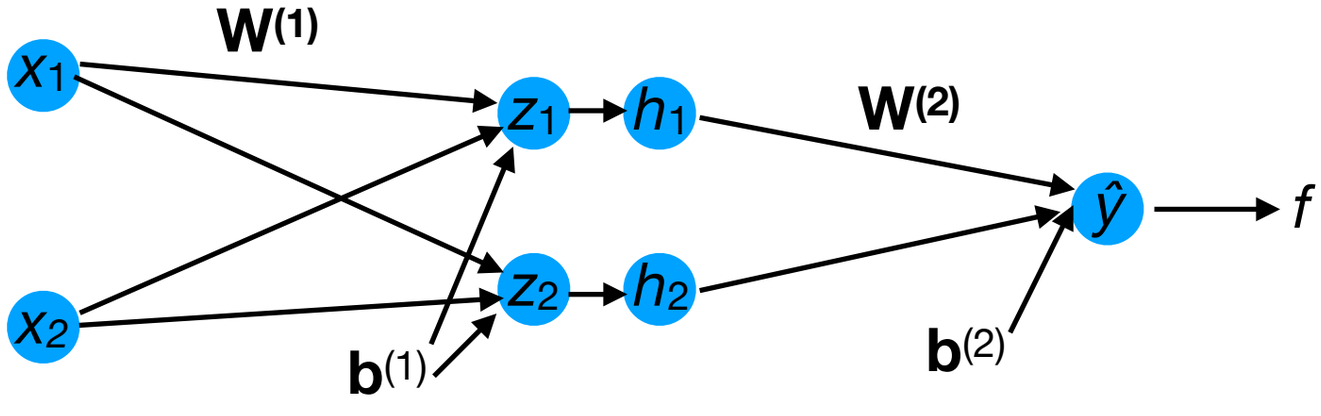
$$\frac{\partial f}{\partial \mathbf{W}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{b}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{b}^{(2)}}$$

Redundant computation

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

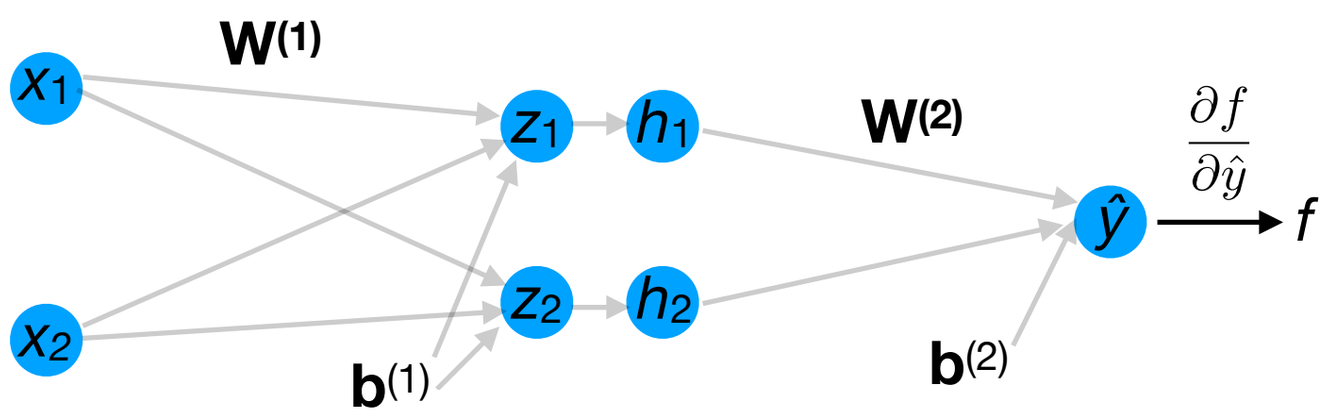
$$\frac{\partial f}{\partial \mathbf{b}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{b}^{(1)}}$$



Computing the gradients

- Here's how we can compute all these *efficiently*:

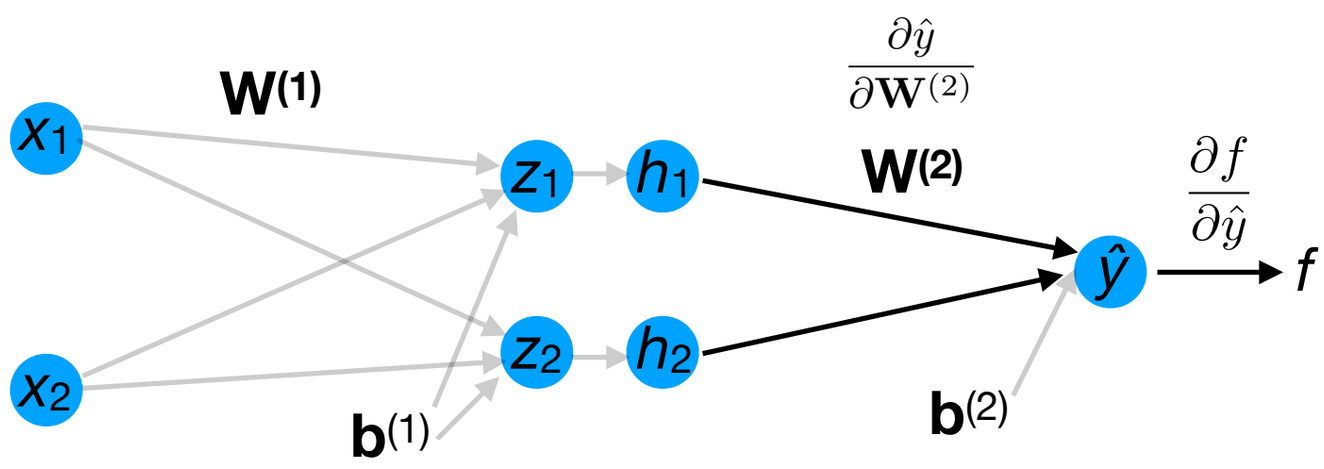
$$\frac{\partial f}{\partial \mathbf{W}^{(2)}} = \frac{\partial f}{\partial \hat{y}}$$



Computing the gradients

- Here's how we can compute all these *efficiently*:

$$\frac{\partial f}{\partial \mathbf{W}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}^{(2)}}$$

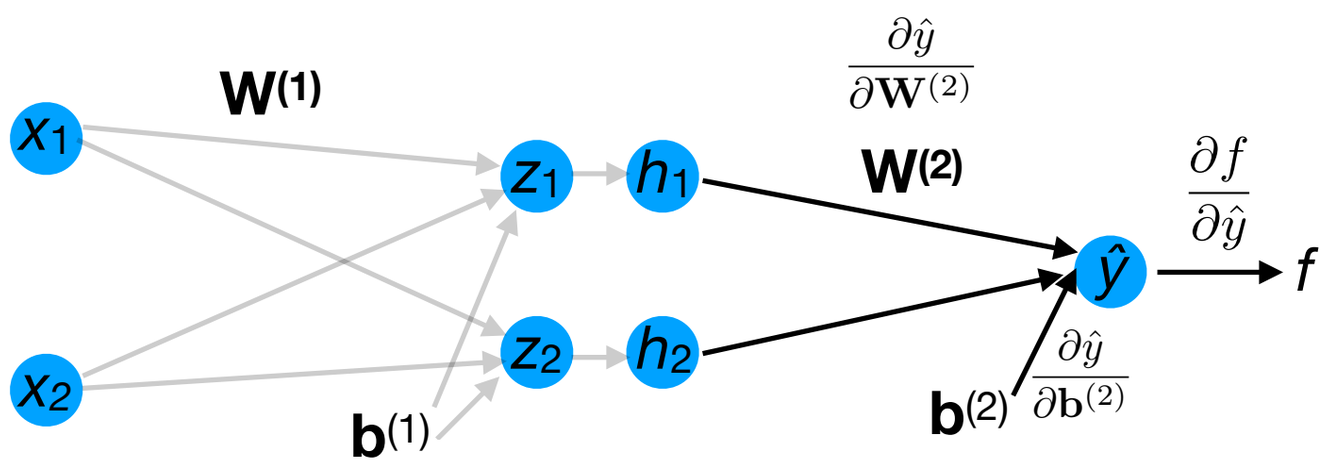


Computing the gradients

- Here's how we can compute all these *efficiently*:

$$\frac{\partial f}{\partial \mathbf{W}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{b}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{b}^{(2)}}$$



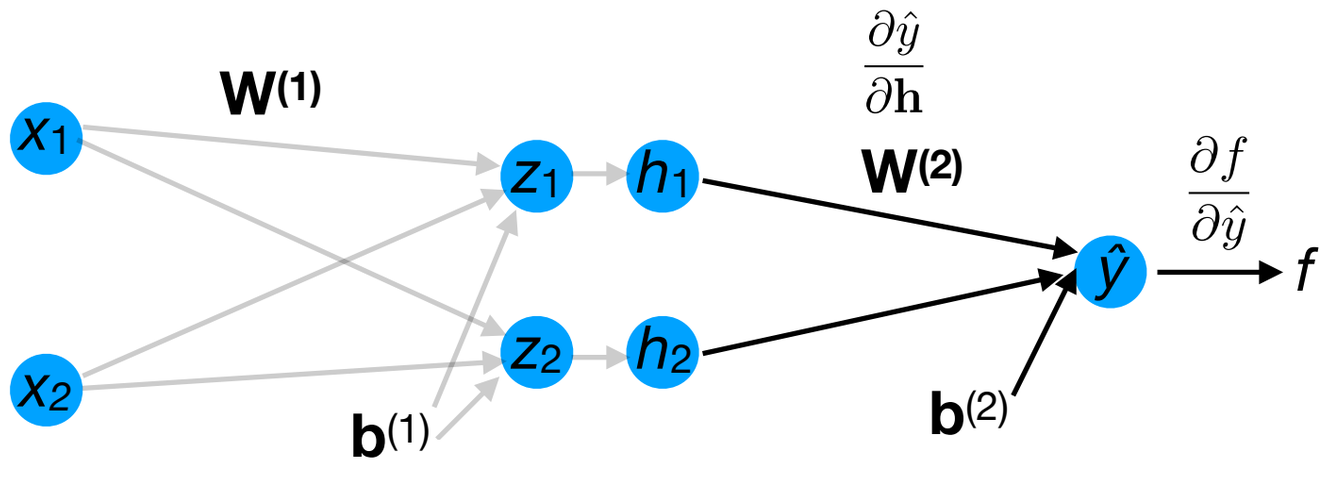
Computing the gradients

- Here's how we can compute all these *efficiently*:

$$\frac{\partial f}{\partial \mathbf{W}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{b}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{b}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}}$$



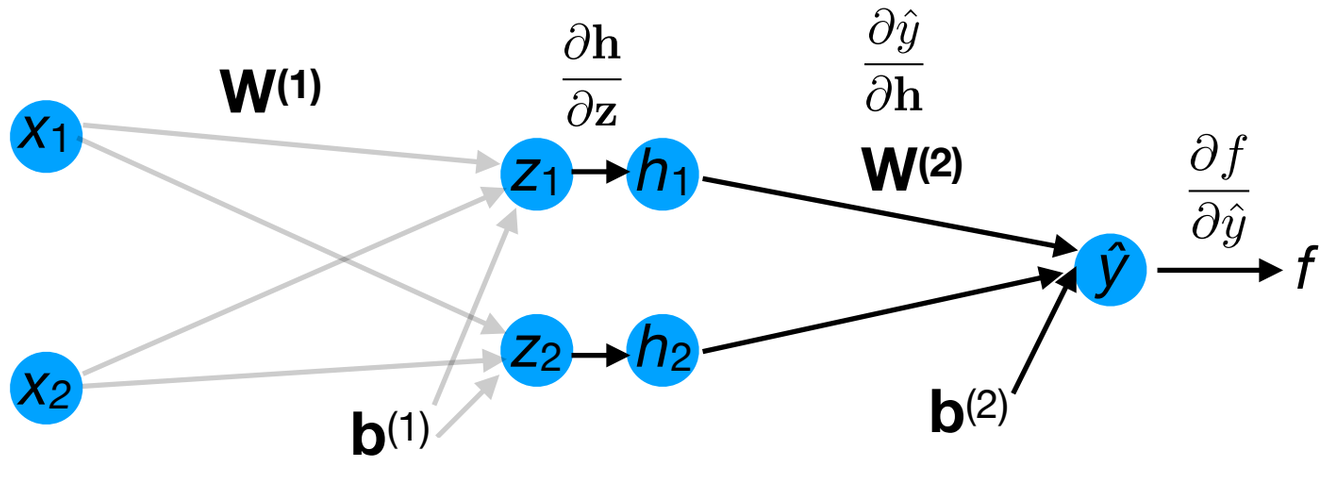
Computing the gradients

- Here's how we can compute all these *efficiently*:

$$\frac{\partial f}{\partial \mathbf{W}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{b}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{b}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}}$$



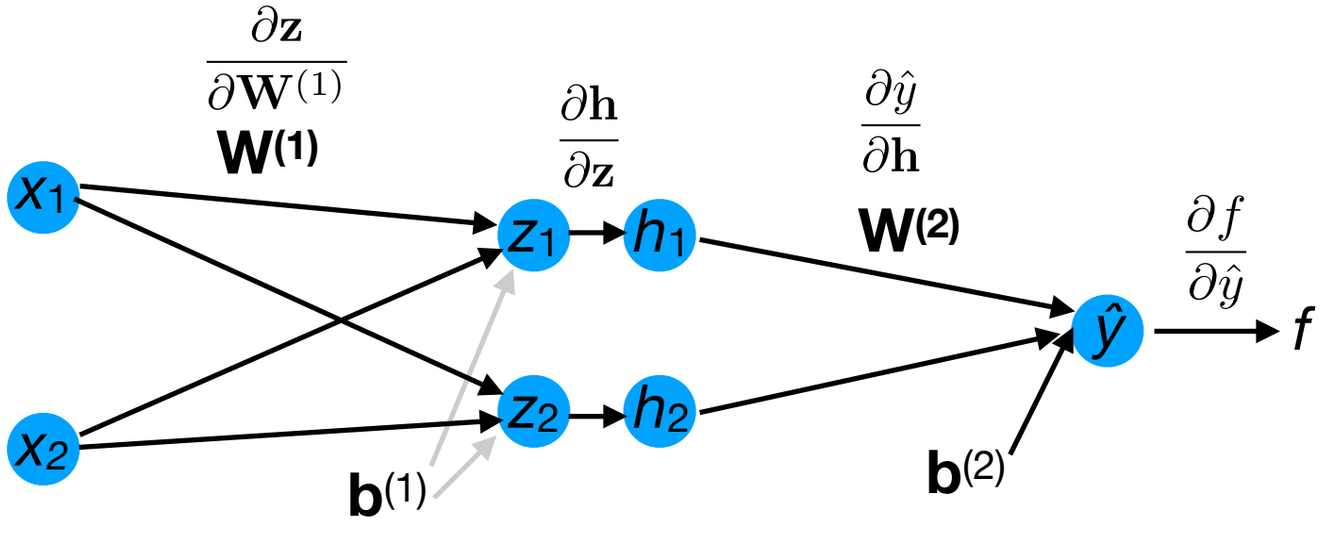
Computing the gradients

- Here's how we can compute all these *efficiently*:

$$\frac{\partial f}{\partial \mathbf{W}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{W}^{(2)}}$$

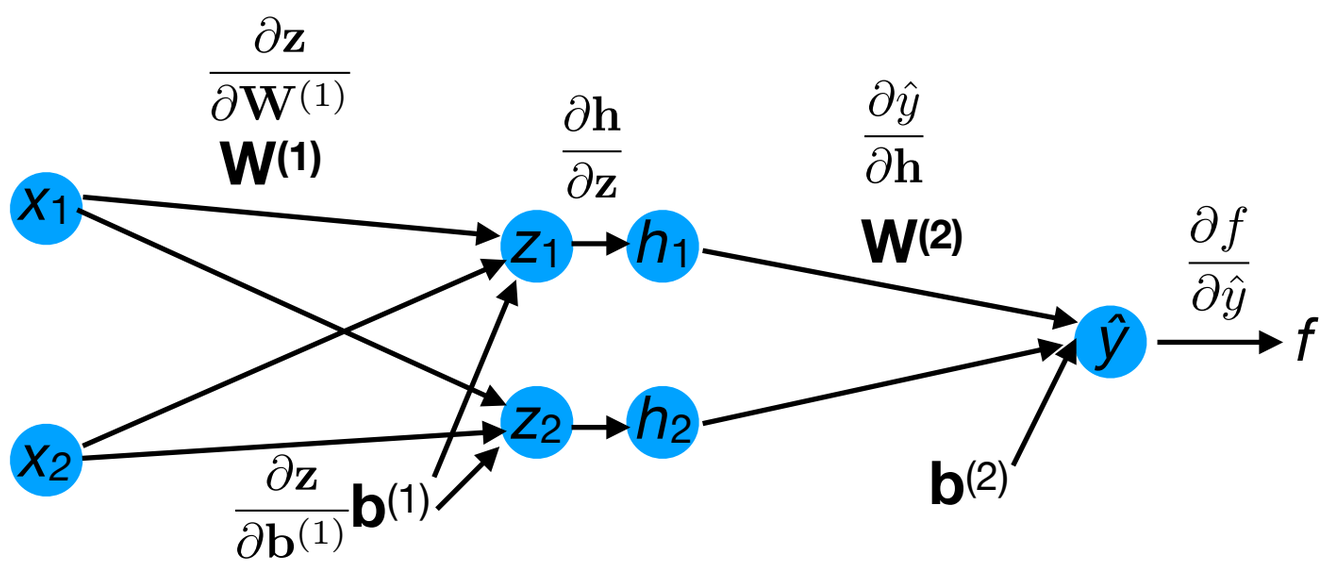
$$\frac{\partial f}{\partial \mathbf{b}^{(2)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{b}^{(2)}}$$

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$



Computing the gradients

- This process is known as **backwards propagation ("backprop"):**
  - It produces the gradient terms of all the weight matrices and bias vectors.



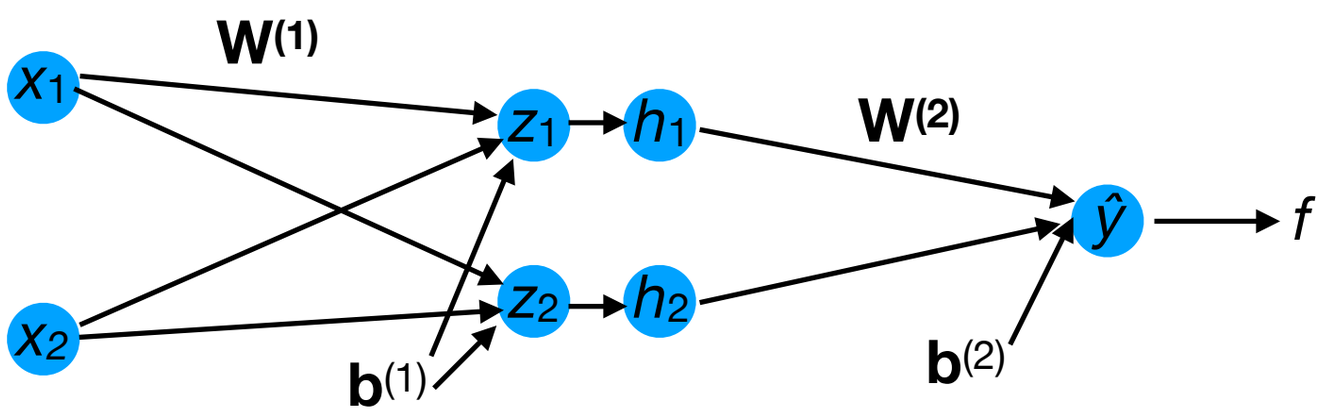
Computing the gradients

- Where do these come from?

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{h}^{(1)\top} \\
\nabla_{\mathbf{b}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}\mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}
\end{aligned}
$$

where

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}}^{\top} - \mathbf{y}) \mathbf{W}^{(2)} \right) \odot \mathrm{relu}'(\mathbf{z}^{(1)\top})
$$



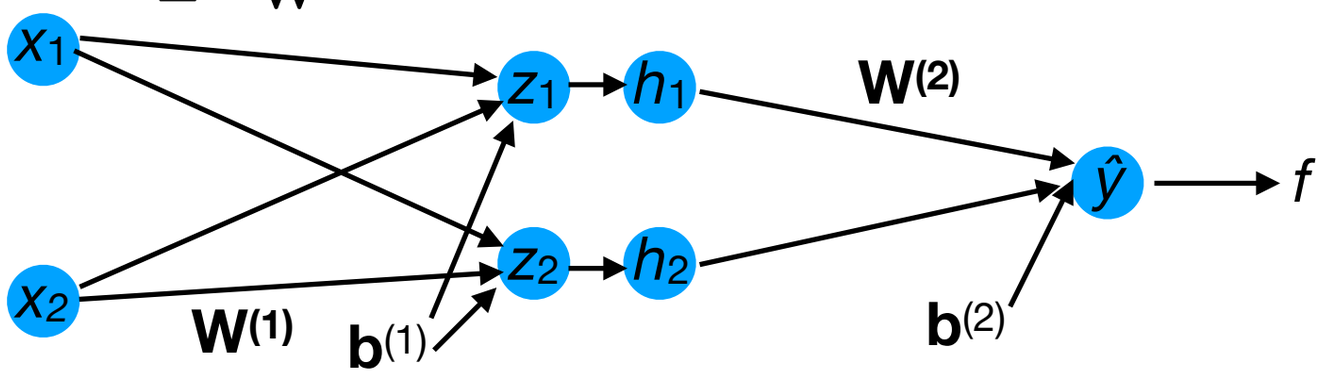
Computing the gradients

- Let's derive each gradient term in turn:

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

- How does $\hat{y}$ depend on $\mathbf{h}$?

$$\begin{aligned} \hat{y} &= \mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)} \\ &= \mathbf{W}_1^{(2)}\mathbf{h}_1 + \mathbf{W}_2^{(2)}\mathbf{h}_2 + \mathbf{b}^{(2)} \\ \implies \frac{\partial \hat{y}}{\partial \mathbf{h}} &= \begin{bmatrix} \frac{\partial \hat{y}}{\partial \mathbf{h}_1} & \frac{\partial \hat{y}}{\partial \mathbf{h}_2} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{W}_1^{(2)} & \mathbf{W}_2^{(2)} \end{bmatrix} \\ &= \mathbf{W}^{(2)} \end{aligned}$$



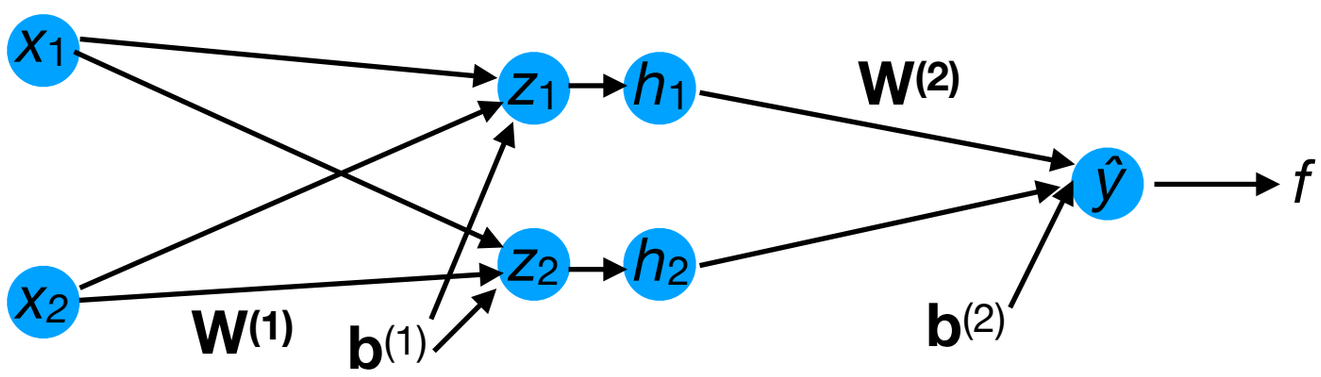
Computing the gradients

- Let's derive each gradient term in turn:

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

- How does **h** depend on **z**?

$$\mathbf{h} = \begin{bmatrix} \text{relu}(\mathbf{z}_1) \\ \text{relu}(\mathbf{z}_2) \end{bmatrix}$$



Computing the gradients

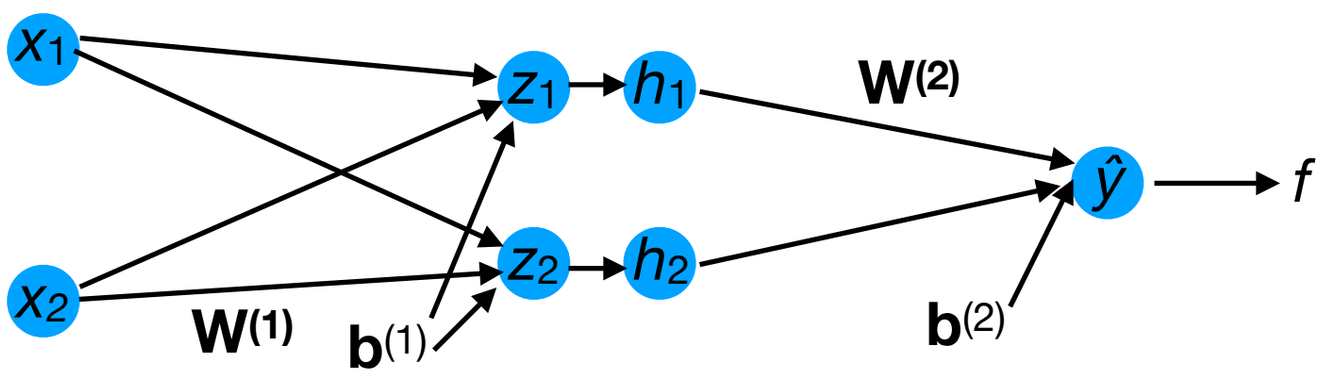
- Let's derive each gradient term in turn:

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

- How does **h** depend on **z**?

$$\mathbf{h} = \begin{bmatrix} \text{relu}(\mathbf{z}_1) \\ \text{relu}(\mathbf{z}_2) \end{bmatrix}$$

$$\implies \frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \begin{bmatrix} \frac{\partial \mathbf{h}_1}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{h}_1}{\partial \mathbf{z}_2} \\ \frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2} \end{bmatrix}$$



Computing the gradients

- Let's derive each gradient term in turn:

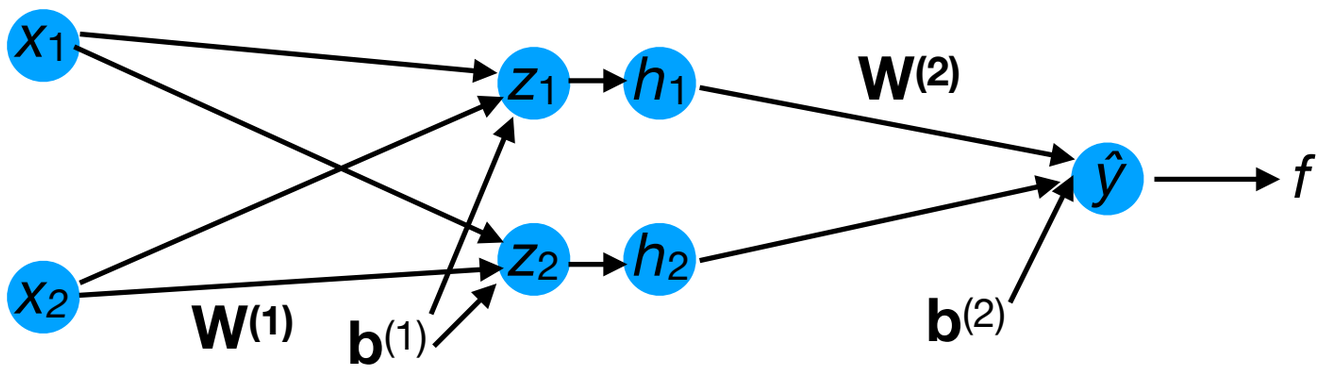
$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

- How does **h** depend on **z**?

$$\mathbf{h} = \begin{bmatrix} \text{relu}(\mathbf{z}_1) \\ \text{relu}(\mathbf{z}_2) \end{bmatrix}$$

$$\implies \frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \begin{bmatrix} \frac{\partial \mathbf{h}_1}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{h}_1}{\partial \mathbf{z}_2} \\ \frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_1} & \frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2} \end{bmatrix}$$

$$= \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & 0 \\ 0 & \text{relu}'(\mathbf{z}_2) \end{bmatrix}$$



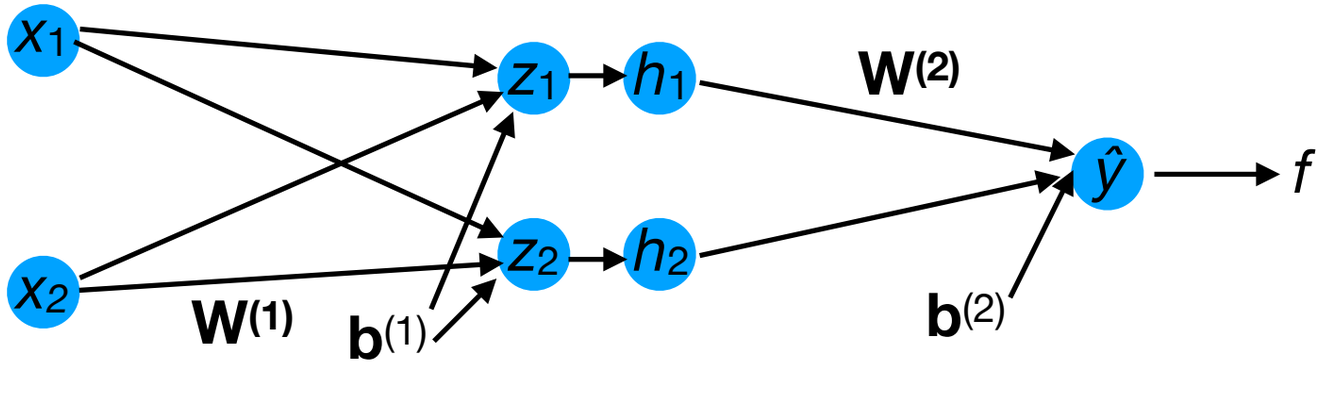
Computing the gradients

- Let's derive each gradient term in turn:

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

- How does **z** depend on **W**(1)?

$$\mathbf{z} = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$$



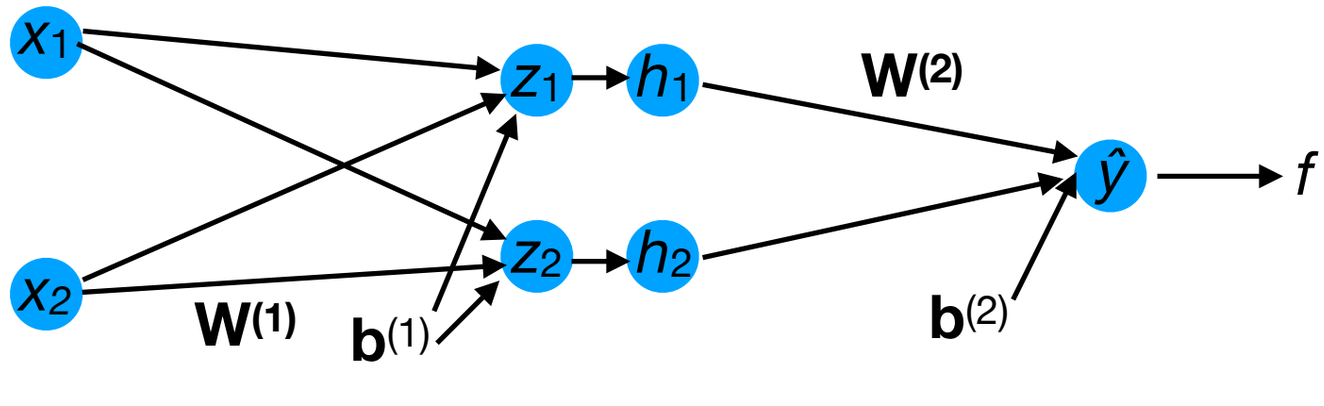
Computing the gradients

- Let's derive each gradient term in turn:

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

- How does **z** depend on **W**(1)?

$$\begin{aligned} \mathbf{z} &= \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \\ \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{W}_1^{(1)} & \mathbf{W}_2^{(1)} \\ \mathbf{W}_3^{(1)} & \mathbf{W}_4^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1^{(1)} \\ \mathbf{b}_2^{(1)} \end{bmatrix} \end{aligned}$$



Computing the gradients

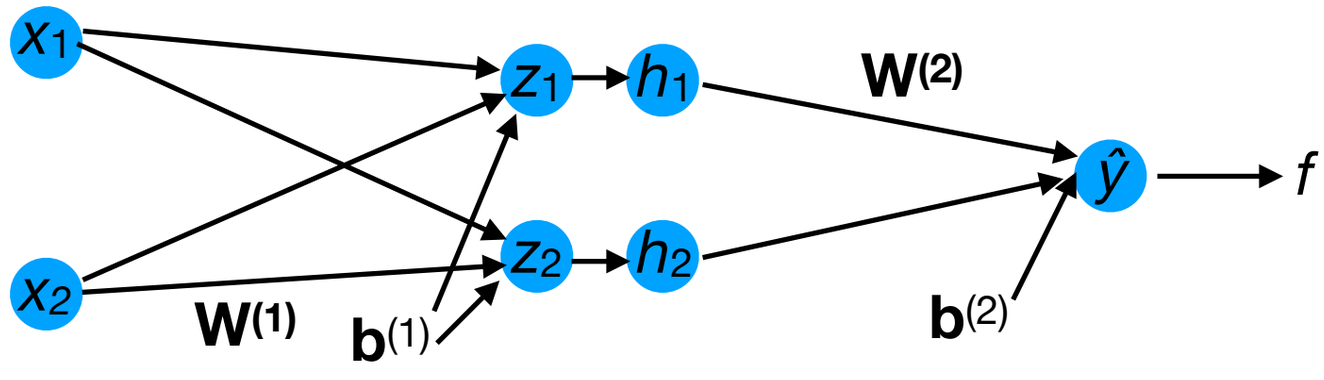
- Let's derive each gradient term in turn:

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

- How does **z** depend on **W**(1)?

$$\begin{aligned}
\mathbf{z} &= \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \\
\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{W}_1^{(1)} & \mathbf{W}_2^{(1)} \\ \mathbf{W}_3^{(1)} & \mathbf{W}_4^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1^{(1)} \\ \mathbf{b}_2^{(1)} \end{bmatrix} \\
\implies \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} &= \begin{bmatrix} \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_1^{(1)}} & \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_2^{(1)}} & \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_3^{(1)}} & \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_4^{(1)}} \\ \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_1^{(1)}} & \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_2^{(1)}} & \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_3^{(1)}} & \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_4^{(1)}} \end{bmatrix}
\end{aligned}$$

For Jacobian matrix, we have to treat W(1) as if it were a *vector*.



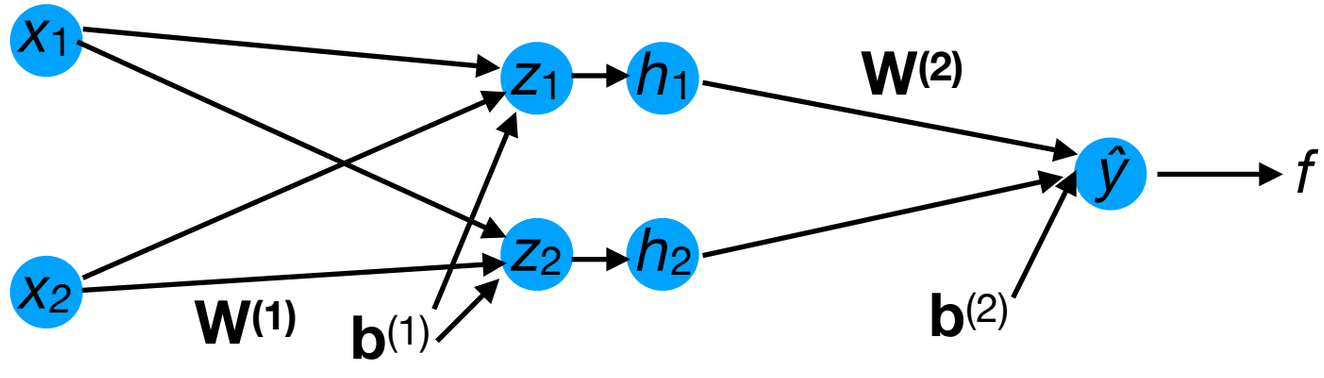
Computing the gradients

- Let's derive each gradient term in turn:

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

- How does **z** depend on **W**(1)?

$$\begin{aligned}
\mathbf{z} &= \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \\
\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{W}_1^{(1)} & \mathbf{W}_2^{(1)} \\ \mathbf{W}_3^{(1)} & \mathbf{W}_4^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1^{(1)} \\ \mathbf{b}_2^{(1)} \end{bmatrix} \\
\implies \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} &= \begin{bmatrix} \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_1^{(1)}} & \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_2^{(1)}} & \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_3^{(1)}} & \frac{\partial \mathbf{z}_1}{\partial \mathbf{W}_4^{(1)}} \\ \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_1^{(1)}} & \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_2^{(1)}} & \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_3^{(1)}} & \frac{\partial \mathbf{z}_2}{\partial \mathbf{W}_4^{(1)}} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}
\end{aligned}$$



Analytical simplification

- We can now finally derive the gradient update for $\mathbf{W}^{(1)}$:

$$\frac{\partial f}{\partial \mathbf{W}^{(1)}} = \frac{\partial f}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}}$$

Analytical simplification

- We can now finally derive the gradient update for $\mathbf{W}^{(1)}$:

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{W}^{(1)}} &= \frac{\partial f}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} \\
&= (\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & 0 \\ 0 & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}
\end{aligned}$$

Analytical simplification

- We can now finally derive the gradient update for $\mathbf{W}^{(1)}$:

$$
\begin{aligned}
\frac{\partial f}{\partial \mathbf{W}^{(1)}} &= \frac{\partial f}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} \\
&= (\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & 0 \\ 0 & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \left(\left((\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \right) \odot \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}
\end{aligned}
$$

since multiplying by a diagonal matrix is equivalent to element-wise (Hadamard) product.

Analytical simplification

- We can now finally derive the gradient update for $\mathbf{W}^{(1)}$:

$$
\begin{aligned}
\frac{\partial f}{\partial \mathbf{W}^{(1)}} &= \frac{\partial f}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} \\
&= (\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & 0 \\ 0 & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \left(\left((\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \right) \odot \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}
\end{aligned}
$$

To simplify notation, let's define a new vector that equals the first few terms.

Analytical simplification

- We can now finally derive the gradient update for $\mathbf{W}^{(1)}$:

$$
\begin{aligned}
\frac{\partial f}{\partial \mathbf{W}^{(1)}} &= \frac{\partial f}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} \\
&= (\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & 0 \\ 0 & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \left(\left((\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \right) \odot \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{g}_1 \mathbf{x}_1 & \mathbf{g}_1 \mathbf{x}_2 & \mathbf{g}_2 \mathbf{x}_1 & \mathbf{g}_2 \mathbf{x}_2 \end{bmatrix}
\end{aligned}
$$

Analytical simplification

- We can now finally derive the gradient update for $\mathbf{W}^{(1)}$:

$$
\begin{aligned}
\frac{\partial f}{\partial \mathbf{W}^{(1)}} &= \frac{\partial f}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{W}^{(1)}} \\
&= (\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & 0 \\ 0 & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \left(\left((\hat{\mathbf{y}} - \mathbf{y})^\top \mathbf{W}^{(2)} \right) \odot \begin{bmatrix} \text{relu}'(\mathbf{z}_1) & \text{relu}'(\mathbf{z}_2) \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & 0 & 0 \\ 0 & 0 & \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{g}_1 \mathbf{x}_1 & \mathbf{g}_1 \mathbf{x}_2 & \mathbf{g}_2 \mathbf{x}_1 & \mathbf{g}_2 \mathbf{x}_2 \end{bmatrix} \\
\implies \nabla_{\mathbf{W}^{(1)}} f &= \mathbf{g}\mathbf{x}^\top
\end{aligned}
$$

Outer product

Weight initialization

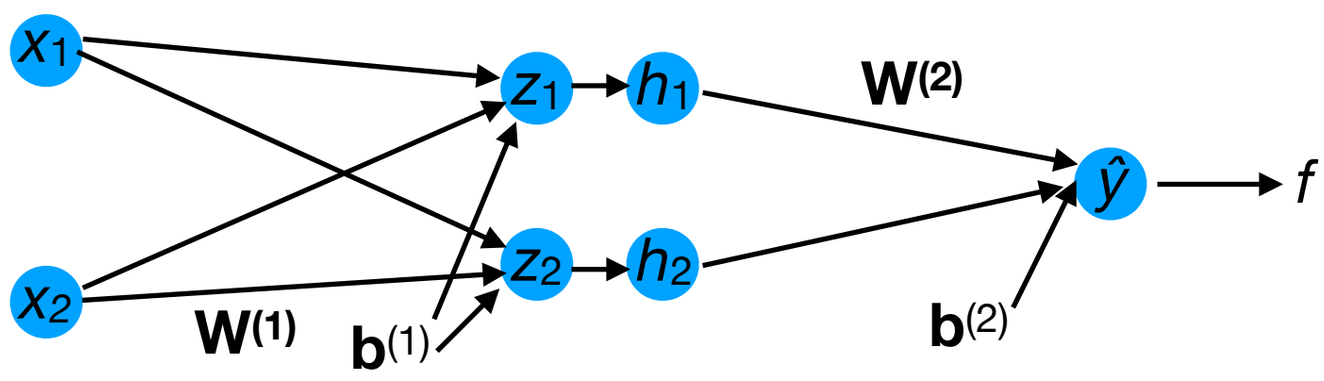
NNs and convexity

- Neural networks are a non-convex ML model.
- Hence, the values you use to initialize the weights and bias terms can make a big difference on the accuracy of the network.
    - The network might end up in a worse local minimum of the cost function.

Weight initialization: example

- Suppose we initialize all the weights and bias terms of a 3-layer NN to be 0.

- What will happen during SGD?

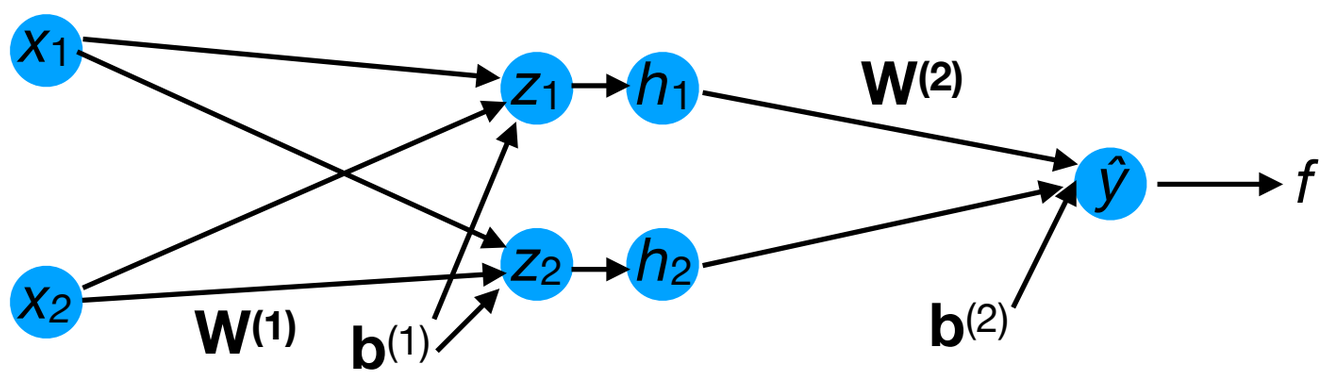


Weight initialization: example

- Suppose we initialize all the weights and bias terms of a 3-layer NN to be 0.

- What will happen during SGD?

During forwards propagation, z and h will be 0. Hence, ŷ will also be 0.



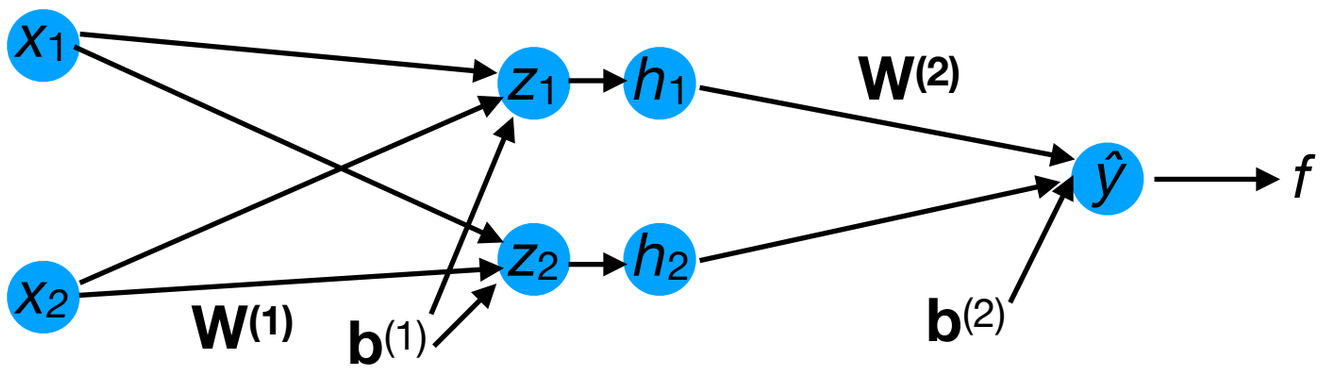
Weight initialization: example

- Suppose we initialize all the weights and bias terms of a 3-layer NN to be 0.

- What will happen during SGD?

During backwards propagation, we have:

$$\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}} = (\hat{\mathbf{y}} - \mathbf{y})\mathbf{h}^{(1)\top}$$
$$\nabla_{\mathbf{b}^{(2)}} f_{\text{CE}} = (\hat{\mathbf{y}} - \mathbf{y})$$
$$\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}} = \mathbf{g}\mathbf{x}^{\top}$$
$$\nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} = \mathbf{g}$$
$$\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top}\mathbf{W}^{(2)}\right) \odot \text{relu}'(\mathbf{z}^{(1)\top})$$



Weight initialization: example

- Suppose we initialize all the weights and bias terms of a 3-layer NN to be 0.

- What will happen during SGD?

During backwards propagation, we have:

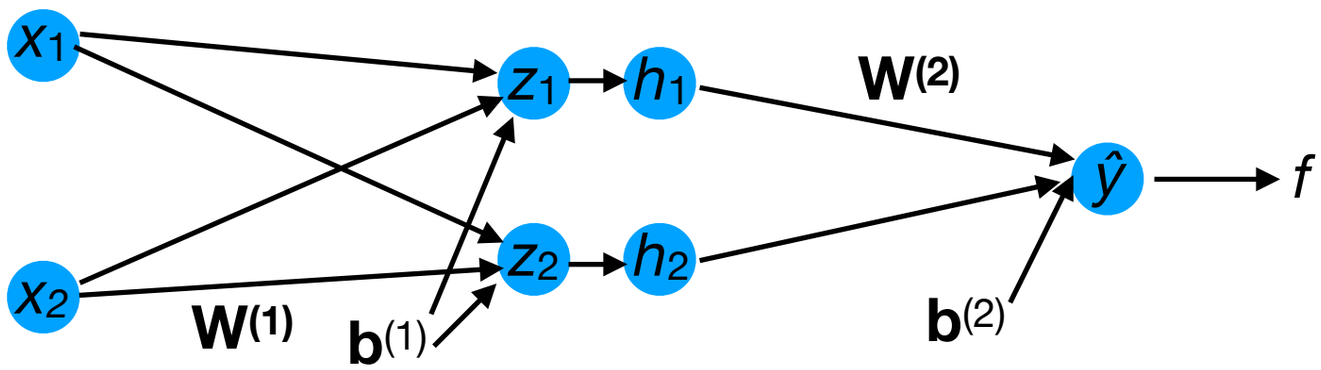
$$\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}} = (\hat{\mathbf{y}} - \mathbf{y})\mathbf{h}^{(1)\top} \; 0$$

$$\nabla_{\mathbf{b}^{(2)}} f_{\text{CE}} = (\hat{\mathbf{y}} - \mathbf{y})$$

$$\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}} = \mathbf{g}\mathbf{x}^{\top} \; 0$$

$$\nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} = \mathbf{g} \; 0$$

$$\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)}\right) \odot \text{relu}'(\mathbf{z}^{(1)\top}) \; 0$$



Weight initialization: example

- Because the gradients w.r.t. $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, and $\mathbf{b}^{(1)}$ are all 0, they will *never change*.

- Only $\mathbf{b}^{(2)}$ will change (to the mean of the target values *y*).

During backwards propagation, we have:

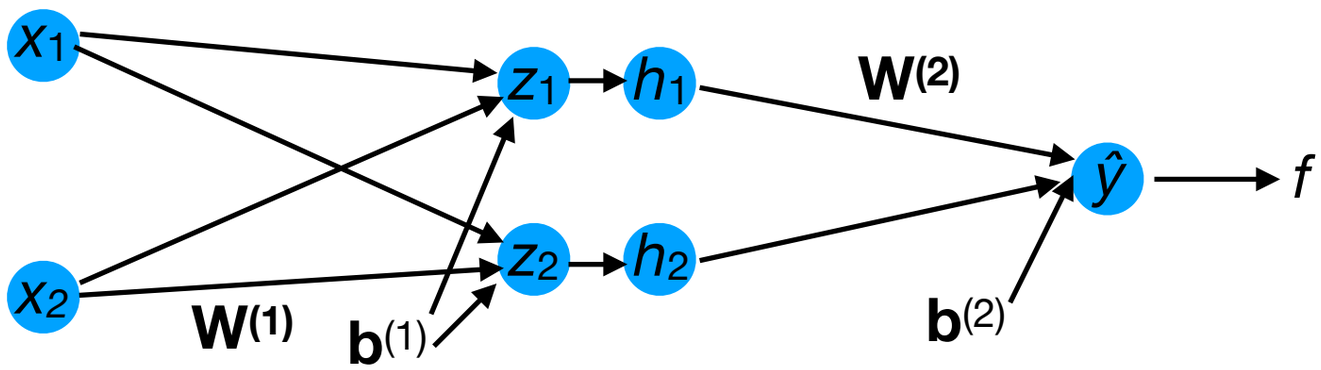
$$\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}} = (\hat{\mathbf{y}} - \mathbf{y})\mathbf{h}^{(1)\top} \; 0$$

$$\nabla_{\mathbf{b}^{(2)}} f_{\text{CE}} = (\hat{\mathbf{y}} - \mathbf{y})$$

$$\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}} = \mathbf{g}\mathbf{x}^{\top} \; 0$$

$$\nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} = \mathbf{g} \; 0$$

$$\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)}\right) \odot \text{relu}'(\mathbf{z}^{(1)\top}) \; 0$$

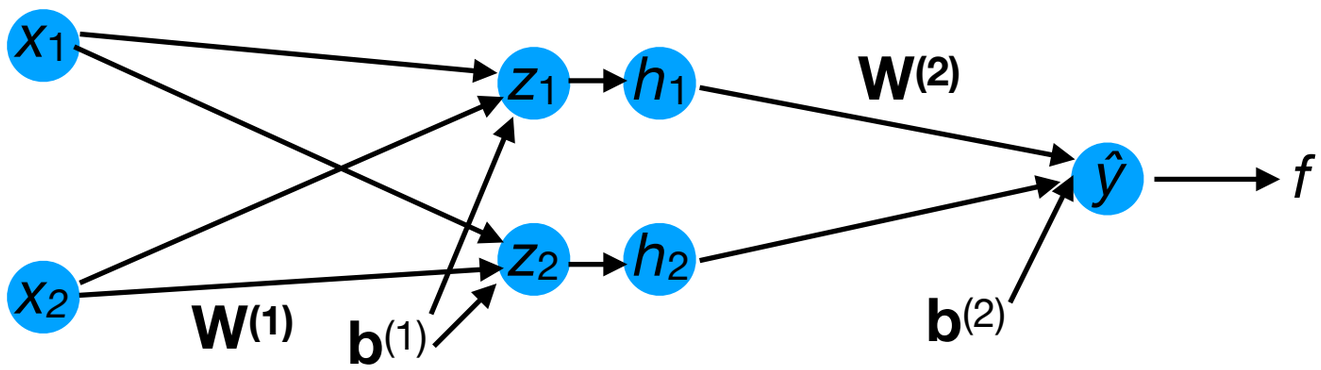


Weight initialization: exercise 1

- Suppose we initialize $\mathbf{W}^{(1)}=\mathbf{b}^{(1)}=0$, but $\mathbf{W}^{(2)}$, $\mathbf{b}^{(2)}$ are non-zero.
- Assume relu'(0)=0.
- What will happen during SGD?

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y})\mathbf{h}^{(1)\top} \\
\nabla_{\mathbf{b}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}\mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)} \right) \odot \mathrm{relu}'(\mathbf{z}^{(1)\top})
$$

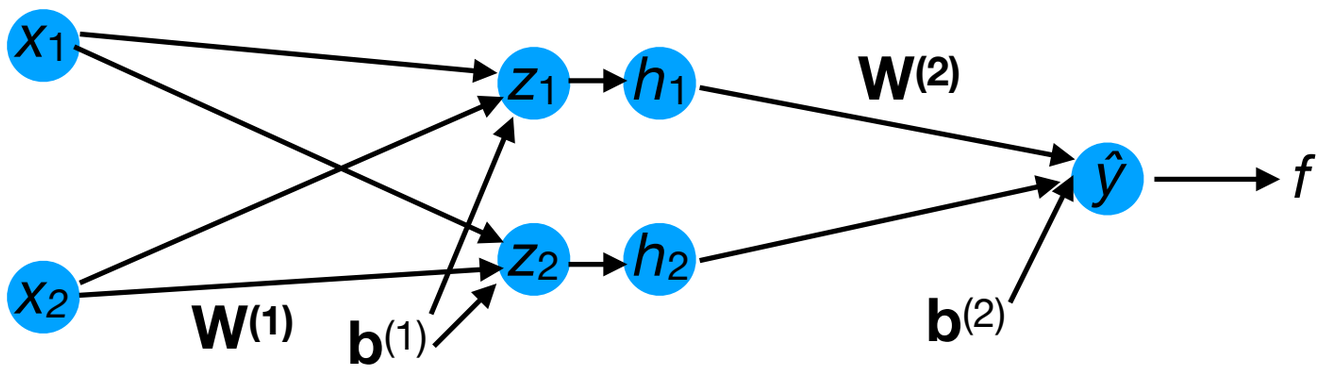


Weight initialization: exercise 1

- Since $\mathbf{W}^{(1)}=\mathbf{b}^{(1)}=0$, then $\mathbf{z}^{(1)}=\mathbf{h}^{(1)}=0$. Hence, $\nabla_{\mathbf{W}^{(2)}} f_{\mathrm{CE}} = 0$.

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}){\mathbf{h}^{(1)}}^{\top} \; \mathbf{0} \\
\nabla_{\mathbf{b}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}\mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)} \right) \odot \mathrm{relu}'({\mathbf{z}^{(1)}}^{\top})
$$

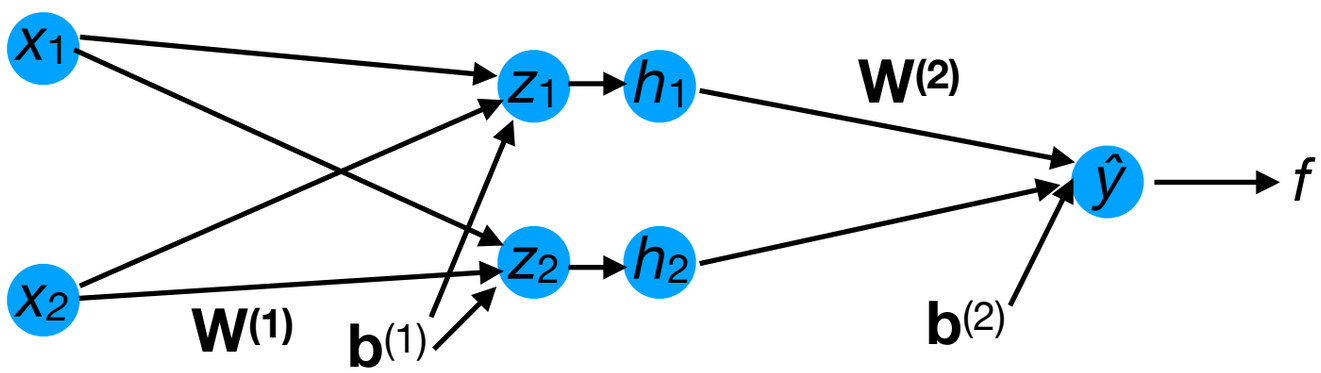


Weight initialization: exercise 1

- Since $\mathbf{W}^{(1)}$=$\mathbf{b}^{(1)}$=0, then $\mathbf{z}^{(1)}$=$\mathbf{h}^{(1)}$=0. Hence, $\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}} = 0$.
- Since relu'(0) = 0, then $\mathbf{g}$=0. Hence, gradients w.r.t. $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$ are 0.
- Only $\mathbf{b}^{(2)}$ can change (so that $\hat{y}$ approaches mean of y).

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{h}^{(1)\top} \quad 0 \\
\nabla_{\mathbf{b}^{(2)}} f_{\text{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}} &= \mathbf{g}\mathbf{x}^{\top} \quad 0 \\
\nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} &= \mathbf{g} \quad 0
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)} \right) \odot \text{relu}'(\mathbf{z}^{(1)\top}) \quad 0
$$

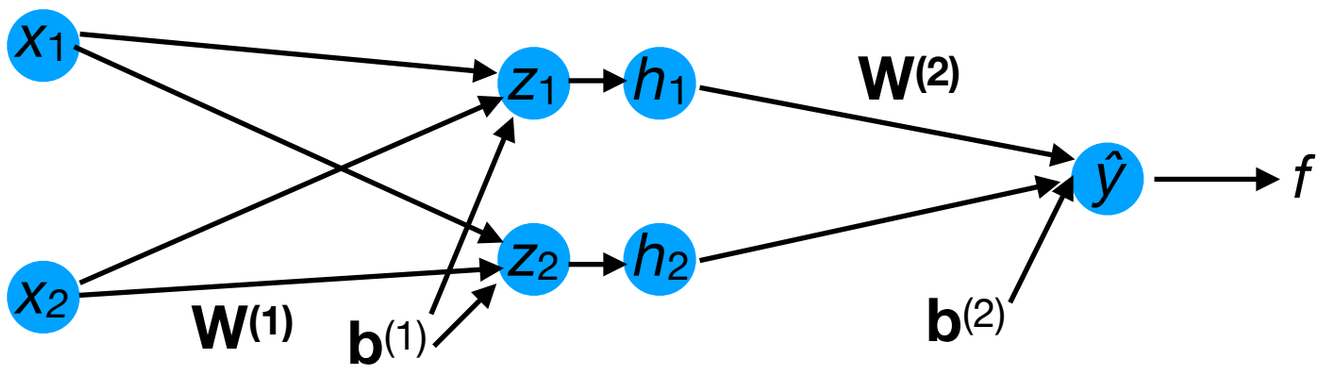


Weight initialization: exercise 2

- Suppose we initialize $\mathbf{W}^{(2)}=\mathbf{b}^{(2)}=0$, but $\mathbf{W}^{(1)}$, $\mathbf{b}^{(1)}$ are non-zero.
- What will happen during SGD?

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}}-\mathbf{y})\mathbf{h}^{(1)\top} \\
\nabla_{\mathbf{b}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}}-\mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}\mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}}-\mathbf{y})^{\top}\mathbf{W}^{(2)}\right) \odot \mathrm{relu}'(\mathbf{z}^{(1)\top})
$$

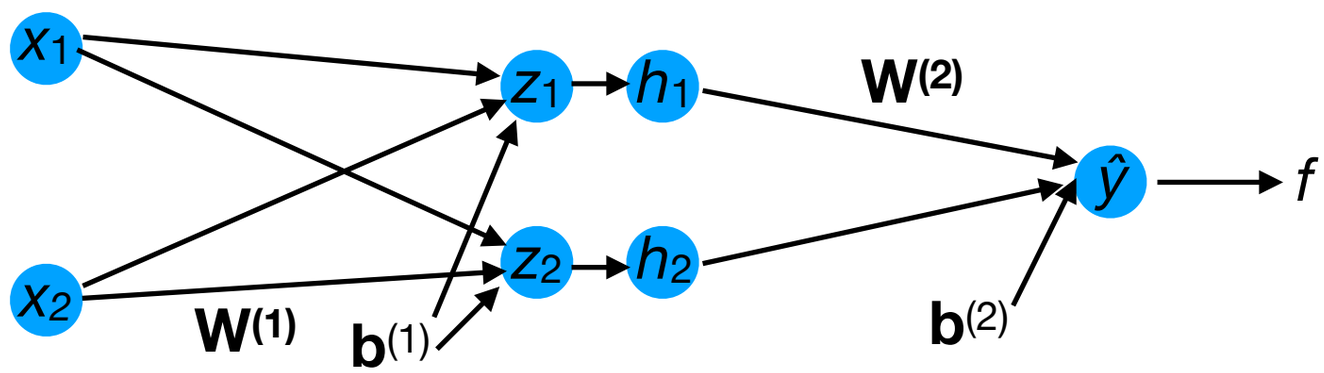


Weight initialization: exercise 2

- Since $\mathbf{W}^{(2)}$=0, then $\mathbf{g}$=0. Hence, $\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}}, \nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} = 0$.

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{h}^{(1)\top} \\
\nabla_{\mathbf{b}^{(2)}} f_{\text{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}} &= \mathbf{g}\mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} &= \mathbf{g}
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)} \right) \odot \text{relu}'(\mathbf{z}^{(1)\top})
$$

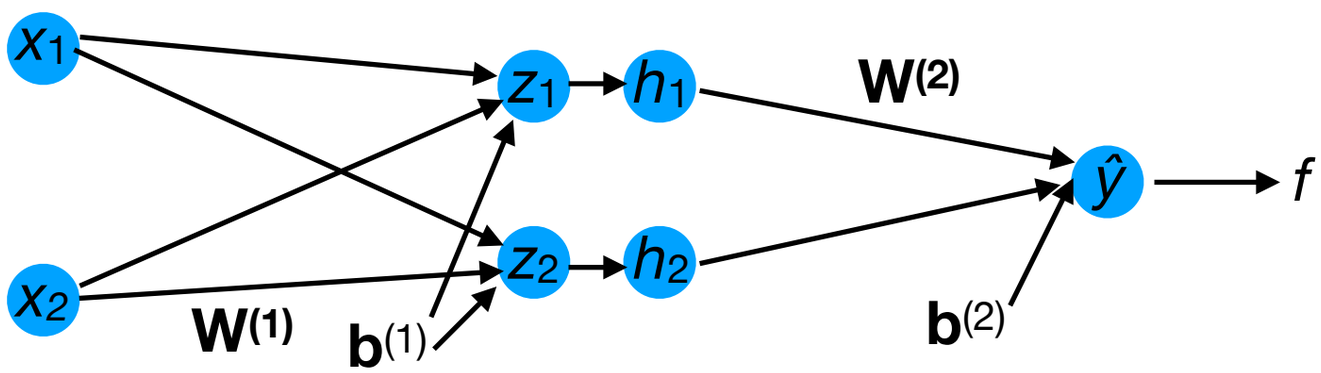


Weight initialization: exercise 2

- Since $\mathbf{W}^{(2)}$=0, then $\mathbf{g}$=0. Hence, $\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}}, \nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} = 0$.
- However, $\mathbf{h}$ is non-zero. Hence, $\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}}$ is nonzero => $\mathbf{W}^{(2)}$ will change.

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{h}^{(1)\top} \\
\nabla_{\mathbf{b}^{(2)}} f_{\text{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}} &= \mathbf{g}\mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} &= \mathbf{g}
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)} \right) \odot \text{relu}'(\mathbf{z}^{(1)\top})
$$

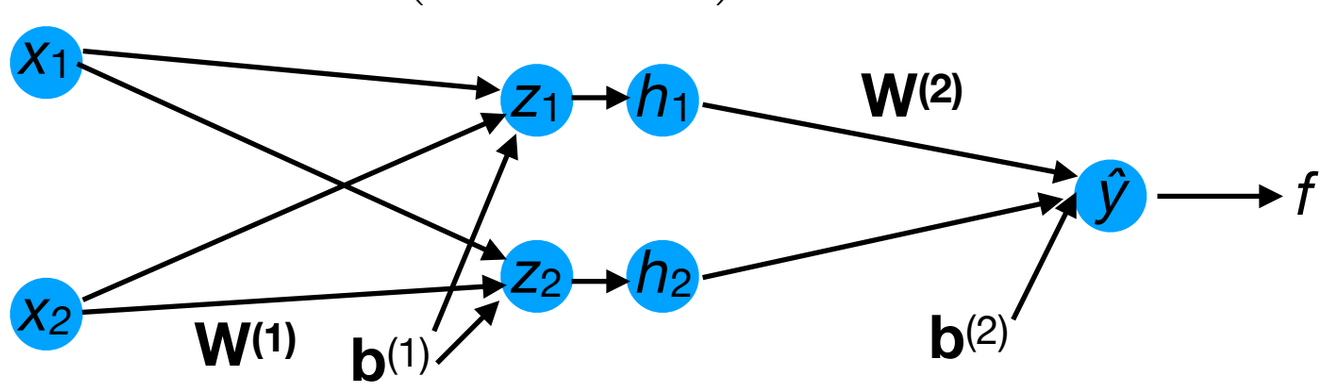


Weight initialization: exercise 2

- Since **W**(2)=0, then **g**=0. Hence, $\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}}, \nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} = 0$.
- However, **h** is non-zero. Hence, $\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}}$ is nonzero => $\mathbf{W}^{(2)}$ will change.
- During the next gradient update, **g** is non-zero => **W**(1), **b**(1) will change.
- In summary: this initialization does not severely inhibit the network's performance (though initializing **W**(2), **b**(2) to 0 is still not recommended).

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\text{CE}} &= (\hat{\mathbf{y}} - \mathbf{y})\mathbf{h}^{(1)\top} \\
\nabla_{\mathbf{b}^{(2)}} f_{\text{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\text{CE}} &= \mathbf{g}\mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\text{CE}} &= \mathbf{g}
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)}\right) \odot \text{relu}'(\mathbf{z}^{(1)\top})
$$



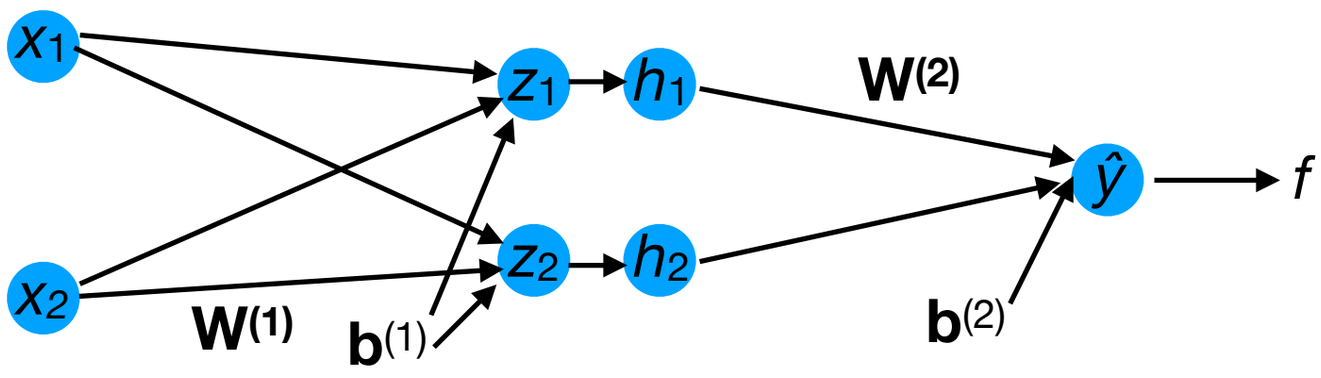
Weight initialization: exercise 3

- Suppose that each weight matrix & bias vector consists of the same row *repeated many times*.

- What will happen during SGD?

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{h}^{(1)\top} \\
\nabla_{\mathbf{b}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g} \mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)} \right) \odot \mathrm{relu}'(\mathbf{z}^{(1)\top})
$$

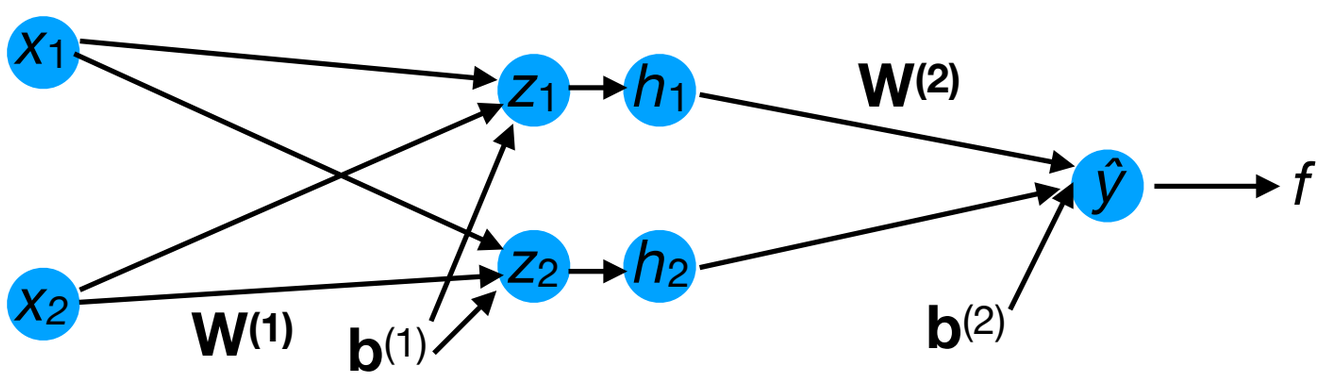


Weight initialization: exercise 3

- In this case, every node of **h** (and **z**) has the *same value*.
- Therefore, the gradient update to each row of $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$ has the *same value*.
- The NN is performing redundant computation — although it has 2 hidden units, it might as well just have 1!

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{h}^{(1)\top} \\
\nabla_{\mathbf{b}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}) \\
\nabla_{\mathbf{W}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g} \mathbf{x}^{\top} \\
\nabla_{\mathbf{b}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}
\end{aligned}
$$

$$
\mathbf{g}^{\top} = \left((\hat{\mathbf{y}} - \mathbf{y})^{\top} \mathbf{W}^{(2)} \right) \odot \mathrm{relu}'(\mathbf{z}^{(1)\top})
$$



Weight initialization methods

- There are various different methods of initializing the weights of a neural network.

- One common approach:
  - For weight matrix $\mathbf{W}^{(j)}$, sample each component from a 0-mean Gaussian with deviation $1/\sqrt{\text{cols}(\mathbf{W}^{(j)})}$.
    - Within certain NNs, helps to ensure that the gradients are usually non-zero.

Weight initialization methods

- There are various different methods of initializing the weights of a neural network.

- One common approach:
  - For weight matrix $\mathbf{W}^{(j)}$, sample each component from a 0-mean Gaussian with deviation $1/\sqrt{\text{cols}(\mathbf{W}^{(j)})}$.
    - Within certain NNs, helps to ensure that the gradients are usually non-zero.
  - *Optional*: orthogonalize the rows of $\mathbf{W}^{(j)}$ to reduce correlation between different units of the pre-activation layer $\mathbf{z}^{(j)}$.

Regularization

L_2 regularization in NNs

- To prevent the weight matrices from growing too big, we can apply an L_2 regularization term to each matrix by augmenting the cross-entropy loss:

$$f_{\mathrm{CE}}(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{10} \mathbf{y}_k^{(i)} \log \hat{\mathbf{y}}_k^{(i)} + \frac{1}{2}\|\mathbf{W}^{(1)}\|_{\mathrm{Fr}}^2 + \frac{1}{2}\|\mathbf{W}^{(2)}\|_{\mathrm{Fr}}^2$$

- Here, $|\mathbf{W}|_{\mathrm{Fr}}^2$ means the squared **Frobenius norm** of **W**.
    - It's just the sum of squares of all the elements of **W**.

L_2 regularization in NNs

- In practice, this just means that the gradients have an additional term, e.g.:

$$
\begin{aligned}
\nabla_{\mathbf{W}^{(2)}} f_{\mathrm{CE}} &= (\hat{\mathbf{y}} - \mathbf{y}){\mathbf{h}^{(1)}}^{\top} + \mathbf{W}^{(2)} \\
\nabla_{\mathbf{W}^{(1)}} f_{\mathrm{CE}} &= \mathbf{g}\mathbf{x}^{\top} + \mathbf{W}^{(1)}
\end{aligned}
$$