DSC 140B

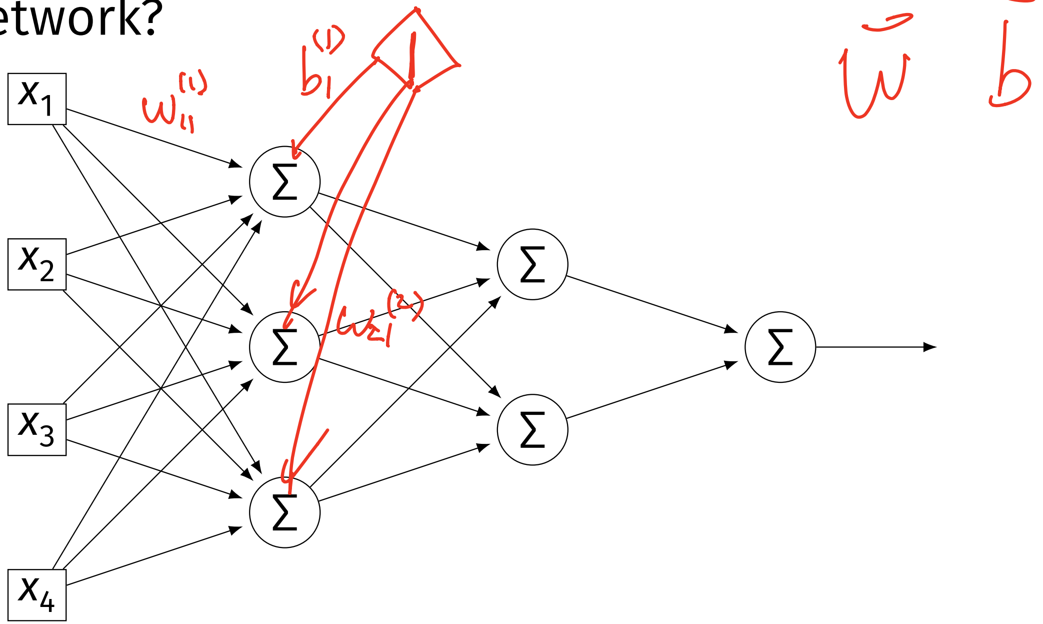
Representation Learning

Lecture 20 | Part 1

Training Neural Networks

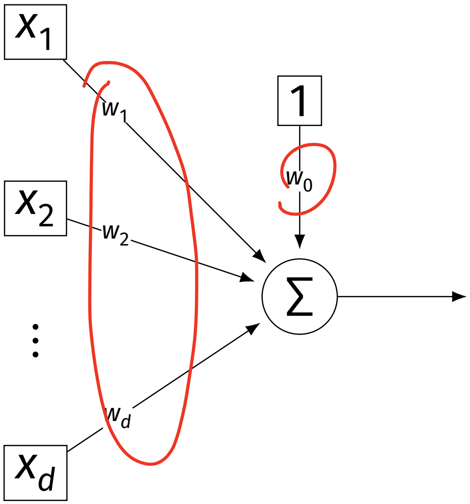
Training

- How do we learn the weights of a (deep) neural network?



Remember...

- How did we learn the weights in linear least squares regression?



Empirical Risk Minimization

0. Collect a training set, $\{(\vec{x}^{(i)}, y_i)\}$
1. Pick the form of the prediction function, H.
2. Pick a loss function.
3. Minimize the empirical risk w.r.t. that loss.

Remember: Linear Least Squares

0. Pick the form of the prediction function, H.
   - E.g., linear: $H(\vec{x};\vec{w}) = w_0 + w_1 x_1 + \ldots + w_d x_d = \text{Aug}(\vec{x}) \cdot \vec{w}$
1. Pick a loss function.
   - E.g., the square loss.
2. Minimize the empirical risk w.r.t. that loss:

$$\min_{\vec{w}} \quad R_{\text{sq}}(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(H(\vec{x}^{(i)}) - y_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\text{Aug}(\vec{x}^{(i)}) \cdot \vec{w} - y_i\right)^2$$

$\frac{\partial R(\vec{w})}{\partial \vec{w}} = 0$ $\vec{w}^* = \ldots$

Minimizing Risk

- To minimize risk, we often use **vector calculus**.
  - Either set $\nabla_{\vec{w}} R(\vec{w}) = 0$ and solve...
  - Or use gradient descent: walk in opposite direction of $\nabla_{\vec{w}} R(\vec{w})$.

$\vec{w}^{(1)}$ $\vec{w}^*$

$\vec{w}^{(0)}$ $\vec{w}^{(2)}$

- Recall, $\nabla_{\vec{w}} R(\vec{w}) = (\partial R/\partial w_0, \partial R/\partial w_1, \ldots, \partial R/\partial w_d)^T$ $\in \mathbb{R}^{k+1}$

$\vec{w} \in \mathbb{R}^{k+1}$

In General

- Let ℓ be the loss function, let $H(\vec{x}; \vec{w})$ be the prediction function.

$$\frac{\partial R}{\partial w} = \frac{\partial R}{\partial H} \frac{\partial H}{\partial \vec{w}}$$

- The empirical risk:

$$R(\vec{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\boxed{H(\vec{x}^{(i)}; \vec{w})}, y_i)$$

- Using the chain rule:

$$\nabla_{\vec{w}} R(\vec{w}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial \ell}{\partial H} \nabla_{\vec{w}} H(\vec{x}^{(i)}; \vec{w})$$

Gradient of H

- To minimize risk, we want to compute $\nabla_{\vec{w}} R$.
- To compute $\nabla_{\vec{w}} R$, we want to compute $\nabla_{\vec{w}} H$.
- This will depend on the form of H.

$\nabla_H R \cdot \nabla_{\vec{w}} H = \nabla_{\vec{w}} R$

Example: Linear Model

- Suppose H is a linear prediction function:

$$H(\vec{x};\vec{w}) = w_0 + w_1 x_1 + \ldots + w_d x_d$$

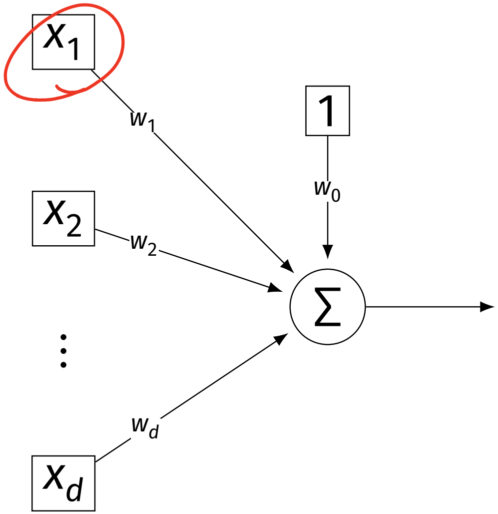
- What is $\nabla_{\vec{w}} H$ with respect to $\vec{w}$?

$$= \left(\nabla_{w_0} H, \nabla_{w_1} H, \ldots, \nabla_{w_d} H \right) \in \mathbb{R}^{d+1}$$

$$= \left(1, x_1, \ldots, x_d \right)$$

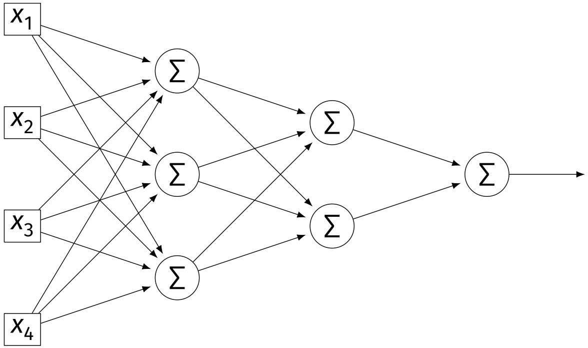
Example: Linear Model

- Consider $\partial H / \partial w_1$: $= x_1$



Example: Neural Networks

- Suppose H is a neural network (with nonlinear activations).
- What is ∇H?
  - It's more complicated...



Parameter Vectors

- It is often useful to pack all of the network's weights into a **parameter vector**, $\vec{w}$.
- Order is arbitrary:

$$\vec{w} = (W_{11}^{(1)}, W_{12}^{(1)}, \ldots, b_1^{(1)}, b_2^{(1)}, W_{11}^{(2)}, W_{12}^{(2)}, \ldots, b_1^{(2)}, b_2^{(2)}, \ldots)^T$$

- The network is a function $H(\vec{x}; \vec{w})$.
- Goal of learning: find the "best" $\vec{w}$.

Gradient of Neural Network

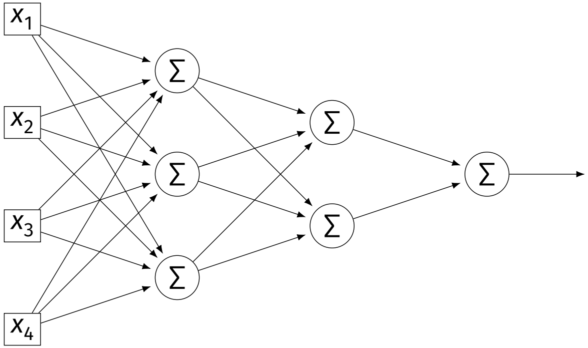
$\vec{w} \in \mathbb{R}^{d'}$

$H(\vec{x}; \vec{w})$

$\nabla_{\vec{w}} H : \vec{x}, \vec{w} \to \mathbb{R}^{d}$

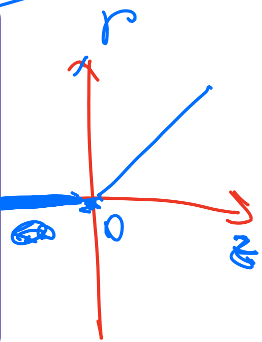
- $\nabla_{\vec{w}} H$ is a vector-valued function.

- Plugging a data point, $\vec{x}$, and a parameter vector, $\vec{w}$, into $\nabla_{\vec{w}} H$ "evaluates the gradient", results in a vector, same size as $\vec{w}$.

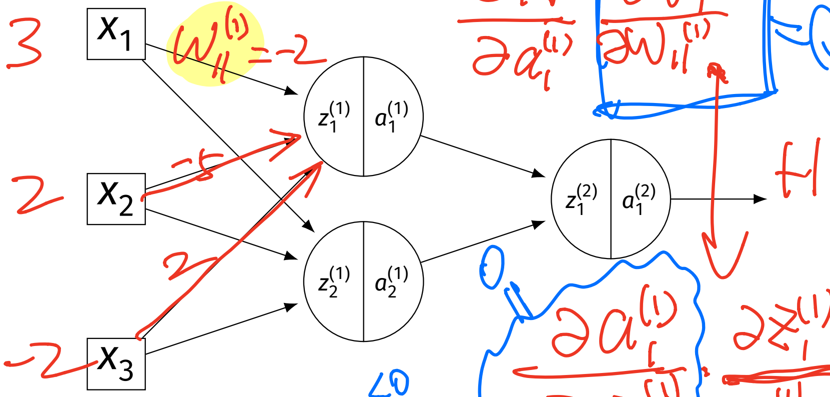


Exercise

Suppose $W^{(1)}_{11} = -2, W^{(1)}_{21} = -5, W^{(1)}_{31} = 2$ and $\vec{x} = (3, 2, -2)^T$ and all biases are 0. ReLU activations are used. What is $\partial H/\partial W^{(1)}_{11}(\vec{x}, \vec{w})$? $= 0$



$= \dfrac{\partial H}{\partial a_1^{(1)}} \dfrac{\partial a_1^{(1)}}{\partial w_{11}^{(1)}}$ $\rightarrow 0$



$0 \quad \dfrac{\partial a_1^{(1)}}{\partial z_1^{(1)}} \cdot \dfrac{\partial z_1^{(1)}}{\partial w_{11}^{(1)}}$

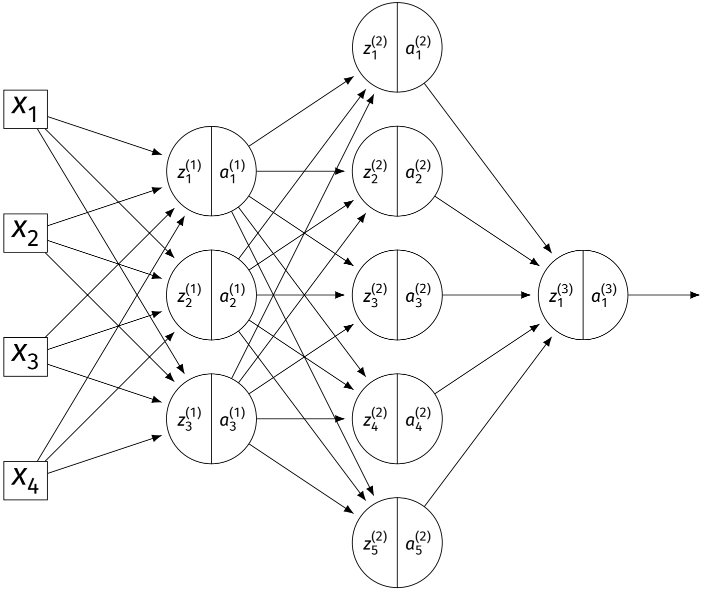
$z_1^{(1)} = 3\times-2 + 2\times-5 + -2\times2$ <0

$\dfrac{\partial r(z)}{\partial z} = 0$

if $z < 0$

Example

- Consider $\partial H / \partial W_{11}^{(1)}$:



A Better Way

- Computing the gradient is straightforward...
- But can involve a lot of repeated work.
- **Backpropagation** is an algorithm for efficiently computing the gradient of a neural network.

DSC 140B
Representation Learning

Lecture 20 | Part 2

Backpropagation

Gradient of a Network

- We want to compute the gradient $\nabla_{\vec{w}} H$.
  - That is, $\partial H / \partial W_{ij}^{(\ell)}$ and $\partial H / \partial b_i^{(\ell)}$ for all valid i, j, ℓ.

- A network is a composition of functions.

- We'll make good use of the **chain rule**.

Recall: The Chain Rule

$$\frac{d}{dx} f(g(x)) = \boxed{\frac{df}{dg}\frac{dg}{dx}}$$

$$= f'(g(x))\, g'(x)$$

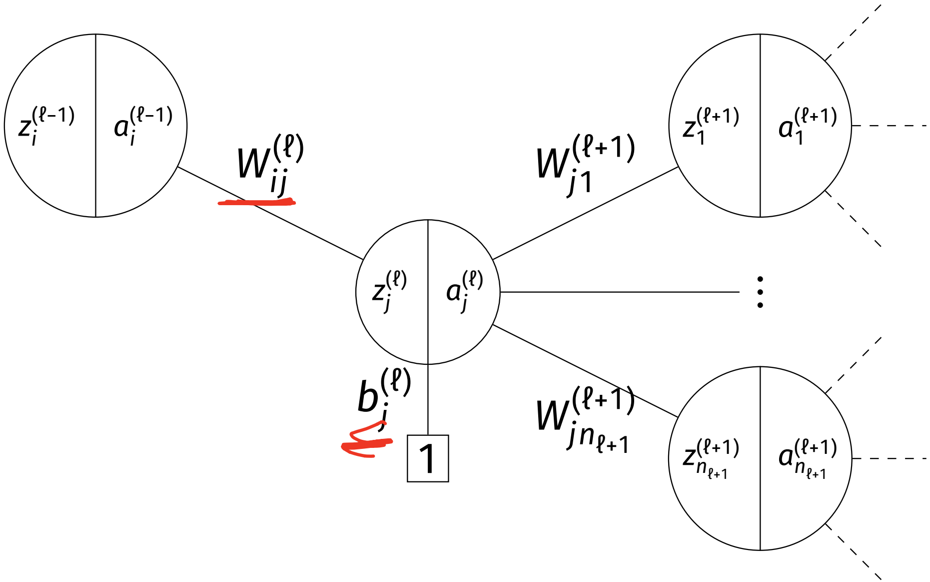
$$\frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x}$$

Some Notation

- We'll consider an arbitrary node in layer ℓ of a neural network.
- Let g be the activation function.
- n_ℓ denotes the number of nodes in layer ℓ.

Arbitrary Node

H



- $\dfrac{\partial H}{\partial W_{ij}^{(\ell)}}$?
- $\dfrac{\partial H}{\partial b_j^{(\ell)}}$?

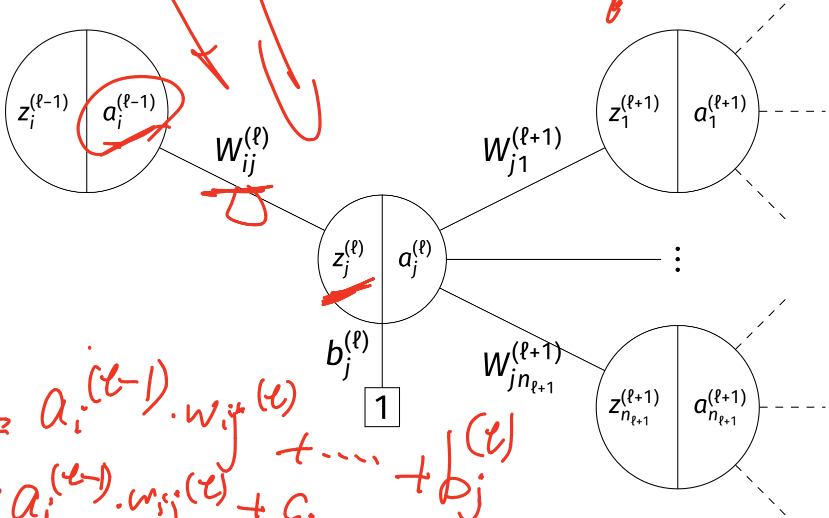
Claim #1

$$H = H\left(z_j^{(\ell)}\right) \qquad z_j^{(\ell)} = ?\left(W_{ij}^{(\ell)}\right)$$

$$\frac{\partial H}{\partial W_{ij}^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}} a_i^{(\ell-1)}$$

$$\frac{\partial H}{\partial w_{ij}^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}} \cdot \frac{\partial z_j^{(\ell)}}{\partial w_{ij}^{(\ell)}}$$

$$a_i^{(\ell-1)}$$



$$z_j^{(\ell)} = a_i^{(\ell-1)} \cdot w_{ij}^{(\ell)} + \cdots + b_j^{(\ell)}$$

$$= a_i^{(\ell-1)} \cdot w_{ij}^{(\ell)} + c.$$

Claim #2

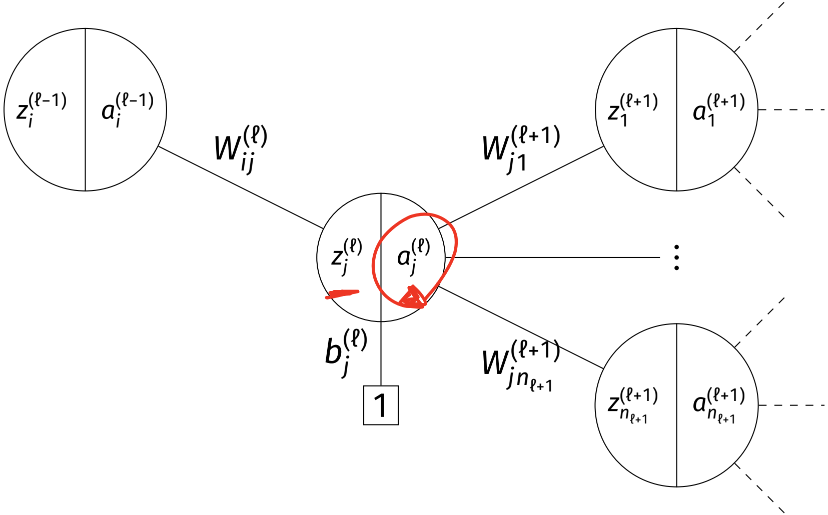
$$\frac{\partial H}{\partial z_j^{(\ell)}} = \frac{\partial H}{\partial a_j^{(\ell)}} g'(z_j^{\ell})$$

$$\frac{\partial H}{\partial z_j^{(\ell)}} = \frac{\partial H}{\partial a_j^{(\ell)}} \cdot \frac{\partial a_j^{(\ell)}}{\partial z_j^{(\ell)}}$$

$$a_j^{(\ell)} = g(z_j^{(\ell)})$$

RELU

$$= \max(0, z_j^{(\ell)})$$



Claim #3

$$f(h(y), q(y))$$

$$\Rightarrow \frac{\partial f}{\partial h} \cdot \frac{\partial h}{\partial y} + \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial y}$$

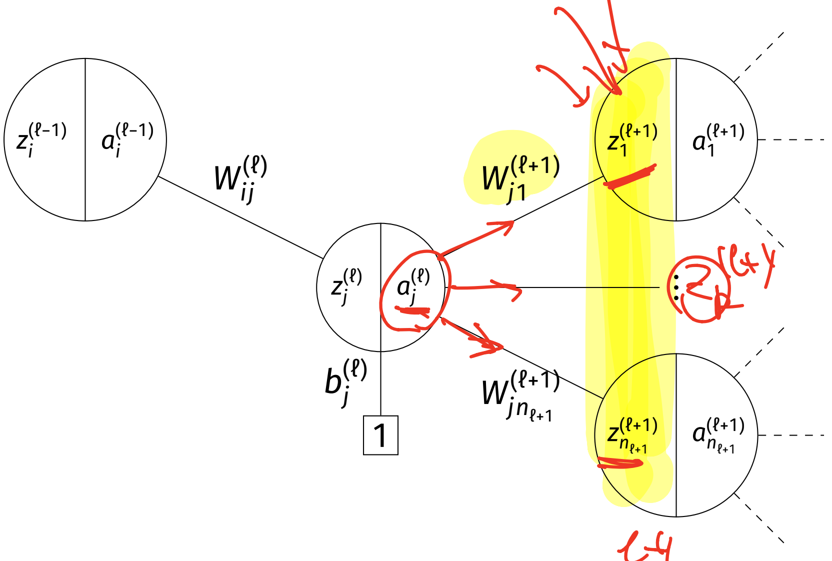
$$\boxed{\frac{\partial H}{\partial a_j^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\partial H}{\partial z_k^{(\ell+1)}} W_{jk}^{(\ell+1)}}$$

$$z_1^{(\ell+1)} = W_{j1}^{(\ell+1)} \cdot a_j^{(\ell)} + C$$

$$\vdots$$

$$z_k^{(\ell+1)} = W_{jk}^{(\ell+1)} \cdot a_j^{(\ell)}$$

$$\vdots$$

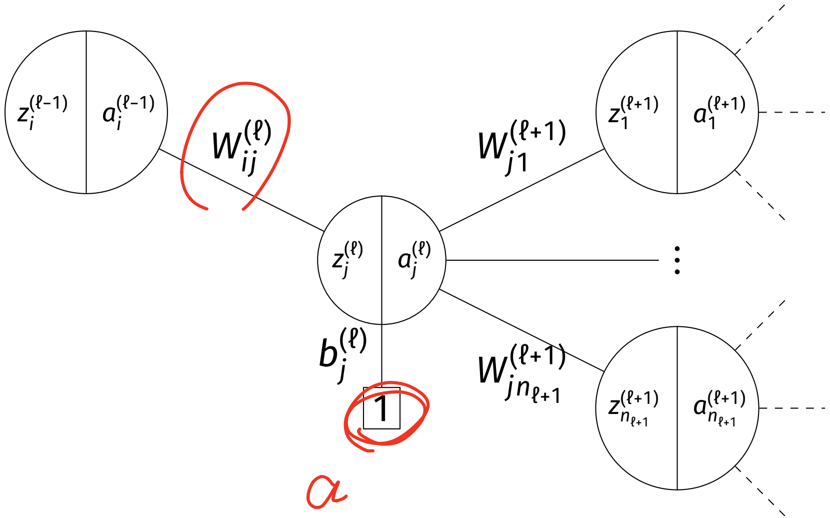


$$H = H\left(z_1^{(\ell+1)}, \ldots, z_k^{(\ell+1)}, \ldots, z_{n_{\ell+1}}^{(\ell+1)}\right)$$

$$\frac{\partial H}{\partial a_j^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\partial H}{\partial z_k^{(\ell+1)}} \cdot \frac{\partial z_k^{(\ell+1)}}{\partial a_j^{(\ell)}}$$

Exercise

What is $\partial H / \partial b_j^{(\ell)}$?



$$\frac{\partial H}{\partial W_{ij}^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}} \cdot a_i^{(\ell-1)}$$

$$\frac{\partial H}{\partial b_j^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}} \cdot 1$$

General Formulas

- For any node in any neural network[1], we have the following recursive formulas:
  - $\dfrac{\partial H}{\partial a_j^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \dfrac{\partial H}{\partial z_k^{(\ell+1)}} W_{jk}^{(\ell+1)}$
  - $\dfrac{\partial H}{\partial z_j^{(\ell)}} = \dfrac{\partial H}{\partial a_j^{(\ell)}} g'(z_j^{\ell})$
  - $\dfrac{\partial H}{\partial W_{ij}^{(\ell)}} = \dfrac{\partial H}{\partial z_j^{(\ell)}} a_i^{(\ell-1)}$
  - $\dfrac{\partial H}{\partial b_j^{(\ell)}} = \dfrac{\partial H}{\partial z_j^{(\ell)}}$

[1]Fully-connected, feedforward network

Main Idea

The derivatives in layer ℓ depend on derivatives in layer $\ell + 1$.

Backpropagation

$\ell = N$

- **Idea:** compute the derivatives in last layers, first.
- That is:
  - Compute derivatives in last layer, ℓ; store them.
  - Use to compute derivatives in layer $\ell - 1$.
  - Use to compute derivatives in layer $\ell - 2$.
  - ...

Backpropagation

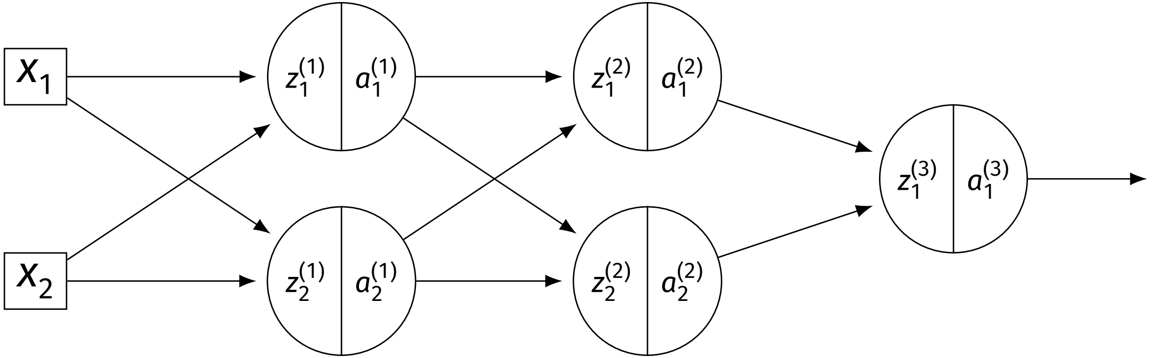
Given an input $\vec{x}$ and a current parameter vector $\vec{w}$:

1. Evaluate the network to compute $z_i^{(\ell)}$ and $a_i^{(\ell)}$ for all nodes.
2. For each layer ℓ from last to first:
   - Compute $\frac{\partial H}{\partial a_j^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\partial H}{\partial z_k^{(\ell+1)}} W_{jk}^{(\ell+1)}$
   - Compute $\frac{\partial H}{\partial z_j^{(\ell)}} = \frac{\partial H}{\partial a_j^{(\ell)}} g'(z_j^{\ell})$
   - Compute $\frac{\partial H}{\partial W_{ij}^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}} a_i^{(\ell-1)}$
   - Compute $\frac{\partial H}{\partial b_j^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}}$

Example

Compute the entries of the gradient given:

$$W^{(1)} = \begin{pmatrix} 2 & -3 \\ 2 & 1 \end{pmatrix} \quad W^{(2)} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \quad W^{(3)} = \begin{pmatrix} 3 \\ -2 \end{pmatrix} \quad \vec{x} = (2,1)^T \quad g(z) = \text{ReLU}$$

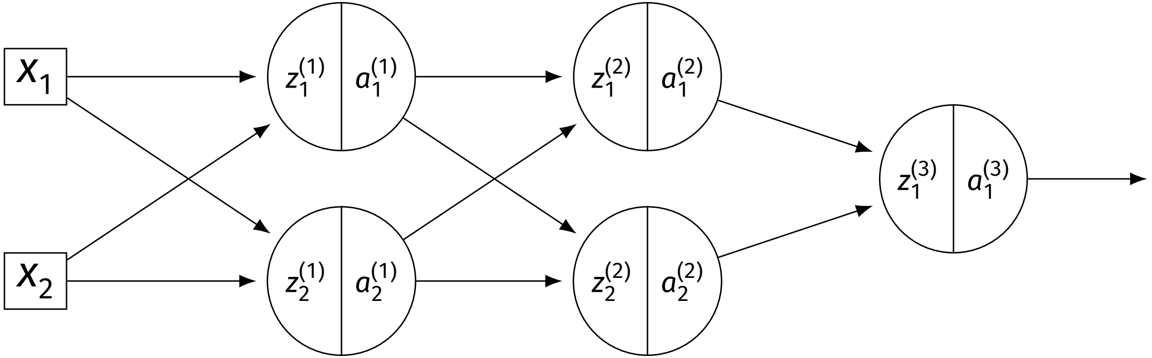


$$\frac{\partial H}{\partial a_j^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\partial H}{\partial z_k^{(\ell+1)}} W_{jk}^{(\ell+1)} \qquad \frac{\partial H}{\partial z_j^{(\ell)}} = \frac{\partial H}{\partial a_j^{(\ell)}} g'(z_j^{\ell}) \qquad \frac{\partial H}{\partial W_{ij}^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}} a_i^{(\ell-1)}$$

Example

Compute the entries of the gradient given:

$W^{(1)} = \begin{pmatrix} 2 & -3 \\ 2 & 1 \end{pmatrix}$ $W^{(2)} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$ $W^{(3)} = \begin{pmatrix} 3 \\ -2 \end{pmatrix}$ $\vec{x} = (2, 1)^T$ $g(z) = \text{ReLU}$

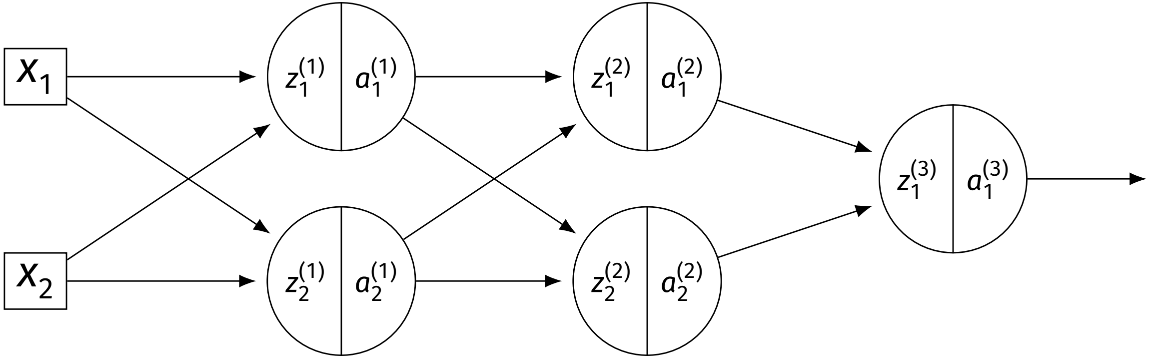


$\frac{\partial H}{\partial a_j^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\partial H}{\partial z_k^{(\ell+1)}} W_{jk}^{(\ell+1)}$ $\frac{\partial H}{\partial z_j^{(\ell)}} = \frac{\partial H}{\partial a_j^{(\ell)}} g'(z_j^{\ell})$ $\frac{\partial H}{\partial W_{ij}^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}} a_i^{(\ell-1)}$

Example

Compute the entries of the gradient given:

$$W^{(1)} = \begin{pmatrix} 2 & -3 \\ 2 & 1 \end{pmatrix} \quad W^{(2)} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \quad W^{(3)} = \begin{pmatrix} 3 \\ -2 \end{pmatrix} \quad \vec{x} = (2, 1)^T \quad g(z) = \text{ReLU}$$



$$\frac{\partial H}{\partial a_j^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\partial H}{\partial z_k^{(\ell+1)}} W_{jk}^{(\ell+1)} \qquad \frac{\partial H}{\partial z_j^{(\ell)}} = \frac{\partial H}{\partial a_j^{(\ell)}} g'(z_j^{\ell}) \qquad \frac{\partial H}{\partial W_{ij}^{(\ell)}} = \frac{\partial H}{\partial z_j^{(\ell)}} a_i^{(\ell-1)}$$

Aside: Derivative of ReLU

$$g(z) = \max\{0, z\}$$

$$g'(z) = \begin{cases} 0, & z < 0 \\ 1, & z > 0 \end{cases}$$