References/Acknowledgments

See the excellent videos by Hugo Larochelle on Backpropagation
Module 4.1: Feedforward Neural Networks (a.k.a. multilayered network of neurons)
The input to the network is an $n$-dimensional vector.
The input to the network is an \( n \)-dimensional vector.
- The input to the network is an \( n \)-dimensional vector
- The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each
- The input to the network is an $n$-dimensional vector
- The network contains $L - 1$ hidden layers (2, in this case) having $n$ neurons each
- The input to the network is an \( n \)-dimensional vector.
- The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each.
- Finally, there is one output layer containing \( k \) neurons (say, corresponding to \( k \) classes).
The input to the network is an \( n \)-dimensional vector.

The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each.

Finally, there is one output layer containing \( k \) neurons (say, corresponding to \( k \) classes).
The input to the network is an $n$-dimensional vector.

The network contains $L - 1$ hidden layers (2, in this case) having $n$ neurons each.

Finally, there is one output layer containing $k$ neurons (say, corresponding to $k$ classes).

Each neuron in the hidden layer and output layer can be split into two parts:

- Pre-activation
- Activation ($a_i$ and $h_i$ are vectors)

The input layer can be called the 0-th layer and the output layer can be called the $(L)$-th layer.

$W_i \in \mathbb{R}^{n \times n}$ and $b_i \in \mathbb{R}^{n}$ are the weight and bias between layers $i-1$ and $i$ ($0 < i < L$).

$W_L \in \mathbb{R}^{n \times k}$ and $b_L \in \mathbb{R}^{k}$ are the weight and bias between the last hidden layer and the output layer ($L = 3$ in this case).
• The input to the network is an $n$-dimensional vector
• The network contains $L - 1$ hidden layers (2, in this case) having $n$ neurons each
• Finally, there is one output layer containing $k$ neurons (say, corresponding to $k$ classes)
• Each neuron in the hidden layer and output layer can be split into two parts:

\[ W_i \in \mathbb{R}^{n \times n} \text{ and } b_i \in \mathbb{R}^n \text{ are the weight and bias between layers } i - 1 \text{ and } i \text{ (} 0 < i < L \text{)} \]
\[ W_L \in \mathbb{R}^{n \times k} \text{ and } b_L \in \mathbb{R}^k \text{ are the weight and bias between the last hidden layer and the output layer } (L = 3 \text{ in this case}) \]
The input to the network is an \( n \)-dimensional vector.

The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each.

Finally, there is one output layer containing \( k \) neurons (say, corresponding to \( k \) classes).

Each neuron in the hidden layer and output layer can be split into two parts: pre-activation.
The input to the network is an \( n \)-dimensional vector.

The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each.

Finally, there is one output layer containing \( k \) neurons (say, corresponding to \( k \) classes).

Each neuron in the hidden layer and output layer can be split into two parts: pre-activation and activation.
The input to the network is an \( n \)-dimensional vector.

The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each.

Finally, there is one output layer containing \( k \) neurons (say, corresponding to \( k \) classes).

Each neuron in the hidden layer and output layer can be split into two parts: pre-activation and activation (\( a_i \) and \( h_i \) are vectors).
The input to the network is an \( \mathbf{n} \)-dimensional vector.

The network contains \( \mathbf{L} - 1 \) hidden layers (2, in this case) having \( \mathbf{n} \) neurons each.

Finally, there is one output layer containing \( \mathbf{k} \) neurons (say, corresponding to \( \mathbf{k} \) classes).

Each neuron in the hidden layer and output layer can be split into two parts: pre-activation and activation (\( a_i \) and \( h_i \) are vectors).

The input layer can be called the 0-th layer and the output layer can be called the \( (L) \)-th layer.
The input to the network is an \( n \)-dimensional vector.

The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each.

Finally, there is one output layer containing \( k \) neurons (say, corresponding to \( k \) classes).

Each neuron in the hidden layer and output layer can be split into two parts: pre-activation and activation (\( a_i \) and \( h_i \) are vectors).

The input layer can be called the 0-th layer and the output layer can be called the \((L)\)-th layer.

\( W_i \in \mathbb{R}^{n \times n} \) and \( b_i \in \mathbb{R}^n \) are the weight and bias between layers \( i - 1 \) and \( i \) \((0 < i < L)\).
- The input to the network is an \( n \)-dimensional vector.
- The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each.
- Finally, there is one output layer containing \( k \) neurons (say, corresponding to \( k \) classes).
- Each neuron in the hidden layer and output layer can be split into two parts: pre-activation and activation (\( a_i \) and \( h_i \) are vectors).
- The input layer can be called the 0-th layer and the output layer can be called the \((L)\)-th layer.
- \( W_i \in \mathbb{R}^{n \times n} \) and \( b_i \in \mathbb{R}^n \) are the weight and bias between layers \( i - 1 \) and \( i \) (\( 0 < i < L \)).
The input to the network is an \( n \)-dimensional vector.

The network contains \( L - 1 \) hidden layers (2, in this case) having \( n \) neurons each.

Finally, there is one output layer containing \( k \) neurons (say, corresponding to \( k \) classes).

Each neuron in the hidden layer and output layer can be split into two parts: pre-activation and activation \((a_i \text{ and } h_i \text{ are vectors})\).

The input layer can be called the 0-th layer and the output layer can be called the \((L)\)-th layer.

\( W_i \in \mathbb{R}^{n \times n} \) and \( b_i \in \mathbb{R}^n \) are the weight and bias between layers \( i - 1 \) and \( i \) \((0 < i < L)\).

\( W_L \in \mathbb{R}^{n \times k} \) and \( b_L \in \mathbb{R}^k \) are the weight and bias between the last hidden layer and the output layer \((L = 3 \text{ in this case})\).
The pre-activation at layer $i$ is given by

$$a_i(x) = b_i + W_i h_{i-1}(x)$$
The pre-activation at layer $i$ is given by

$$a_i(x) = b_i + W_i h_{i-1}(x)$$

The activation at layer $i$ is given by

$$h_i(x) = g(a_i(x))$$
The pre-activation at layer $i$ is given by

$$a_i(x) = b_i + W_i h_{i-1}(x)$$

The activation at layer $i$ is given by

$$h_i(x) = g(a_i(x))$$

where $g$ is called the activation function (for example, logistic, tanh, linear, etc.)
The pre-activation at layer $i$ is given by

$$a_i(x) = b_i + W_i h_{i-1}(x)$$

The activation at layer $i$ is given by

$$h_i(x) = g(a_i(x))$$

where $g$ is called the activation function (for example, logistic, tanh, linear, etc.).

The activation at the output layer is given by

$$f(x) = h_L(x) = O(a_L(x))$$
\[ h_L = \hat{y} = f(x) \]

- The pre-activation at layer \( i \) is given by
  \[ a_i(x) = b_i + W_i h_{i-1}(x) \]
- The activation at layer \( i \) is given by
  \[ h_i(x) = g(a_i(x)) \]
  where \( g \) is called the activation function (for example, logistic, tanh, linear, etc.)
- The activation at the output layer is given by
  \[ f(x) = h_L(x) = O(a_L(x)) \]
  where \( O \) is the output activation function (for example, softmax, linear, etc.)
The pre-activation at layer $i$ is given by

$$a_i(x) = b_i + W_i h_{i-1}(x)$$

The activation at layer $i$ is given by

$$h_i(x) = g(a_i(x))$$

where $g$ is called the activation function (for example, logistic, tanh, linear, etc.)

The activation at the output layer is given by

$$f(x) = h_L(x) = O(a_L(x))$$

where $O$ is the output activation function (for example, softmax, linear, etc.)

To simplify notation we will refer to $a_i(x)$ as $a_i$ and $h_i(x)$ as $h_i$. 

$h_L = \hat{y} = f(x)$
The pre-activation at layer $i$ is given by:

$$a_i = b_i + W_i h_{i-1}$$

The activation at layer $i$ is given by:

$$h_i = g(a_i)$$

where $g$ is called the activation function (for example, logistic, tanh, linear, etc.)

The activation at the output layer is given by:

$$f(x) = h_L = O(a_L)$$

where $O$ is the output activation function (for example, softmax, linear, etc.)
\[ h_L = \hat{y} = f(x) \]

- **Data:** \( \{x_i, y_i\}_{i=1}^N \)

\[ \hat{y}_i = f(x_i) = O(W_3 g(W_2 g(W_1 x_1 + b_1) + b_2) + b_3) \]

**Parameters:** \( \theta = W_1, \ldots, W_L, b_1, b_2, \ldots, b_L \) (\( L = 3 \))

**Algorithm:** Gradient Descent with Back-propagation (we will see soon)

**Objective/Loss/Error function:**

\[ \min_{1 \leq i \leq N} \sum_{j=1}^k (\hat{y}_{ij} - y_{ij})^2 \]

In general, \( \min_L(\theta) \) where \( L(\theta) \) is some function of the parameters.
\[ h_L = \hat{y} = f(x) \]

- **Data:** \( \{x_i, y_i\}_{i=1}^{N} \)
- **Model:**

\[
\hat{y}_i = f(x_i) = O(W_3 g(W_2 g(W_1 x_1 + b_1) + b_2) + b_3)
\]

**Parameters:** \( \theta = W_1, \ldots, W_L, b_1, b_2, \ldots, b_L \) \( (L = 3) \)

**Algorithm:** Gradient Descent with Back-propagation (we will see soon)

**Objective/Loss/Error function:**
Say,
\[
\min \sum_{i=1}^{N} \sum_{j=1}^{k} (\hat{y}_{ij} - y_{ij})^2
\]

In general,
\[
\min L(\theta)
\]
where \( L(\theta) \) is some function of the parameters
\[ h_L = \hat{y} = f(x) \]

- **Data:** \( \{x_i, y_i\}_{i=1}^{N} \)
- **Model:**

\[
\hat{y}_i = f(x_i) = O(W_3g(W_2g(W_1x + b_1) + b_2) + b_3)
\]
\( h_L = \hat{y} = f(x) \)

- **Data:** \( \{x_i, y_i\}_{i=1}^N \)
- **Model:**

\[
\hat{y}_i = f(x_i) = O(W_3g(W_2g(W_1x + b_1) + b_2) + b_3)
\]

- **Parameters:**

\( \theta = W_1, \ldots, W_L, b_1, b_2, \ldots, b_L (L = 3) \)
- **Data:** $\{x_i, y_i\}_{i=1}^N$
- **Model:**
  \[ \hat{y}_i = f(x_i) = O(W_3g(W_2g(W_1x + b_1) + b_2) + b_3) \]
- **Parameters:**
  \[ \theta = W_1, .., W_L, b_1, b_2, ..., b_L (L = 3) \]
- **Algorithm:** Gradient Descent with Backpropagation (we will see soon)
Data: $\{x_i, y_i\}_{i=1}^N$

Model:

$$\hat{y}_i = f(x_i) = O(W_3g(W_2g(W_1x + b_1) + b_2) + b_3)$$

Parameters:

$$\theta = W_1,..,W_L, b_1, b_2, ..., b_L (L = 3)$$

Algorithm: Gradient Descent with Backpropagation (we will see soon)

Objective/Loss/Error function: Say,

$$\min \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} (\hat{y}_{ij} - y_{ij})^2$$

In general, $\min \mathcal{L}(\theta)$

where $\mathcal{L}(\theta)$ is some function of the parameters.
Module 4.2: Learning Parameters of Feedforward Neural Networks (Intuition)
The story so far...

- We have introduced feedforward neural networks
- We are now interested in finding an algorithm for learning the parameters of this model
Recall our gradient descent algorithm.
Recall our gradient descent algorithm

**Algorithm:** gradient_descent()

\[ t \leftarrow 0; \]
\[ max\_iterations \leftarrow 1000; \]
\[ \text{Initialize } w_0, b_0; \]
\[ \textbf{while } t++ < max\_iterations \textbf{ do} \]
\[ \quad w_{t+1} \leftarrow w_t - \eta \nabla w_t; \]
\[ \quad b_{t+1} \leftarrow b_t - \eta \nabla b_t; \]
\[ \textbf{end} \]
$h_L = \hat{y} = f(x)$

- Recall our gradient descent algorithm
- We can write it more concisely as

**Algorithm: gradient_descent()**

\[
t \leftarrow 0;
\]
\[
max\_iterations \leftarrow 1000;
\]
\[
Initialize \ w_0, b_0;
\]
\[
\textbf{while } t++ < max\_iterations \textbf{ do}
\]
\[
\quad w_{t+1} \leftarrow w_t - \eta \nabla w_t;
\]
\[
\quad b_{t+1} \leftarrow b_t - \eta \nabla b_t;
\]
\[
\textbf{end}
\]
Recall our gradient descent algorithm.

We can write it more concisely as

**Algorithm:** gradient_descent()

\[
t \leftarrow 0; \\
max_{iterations} \leftarrow 1000; \\
Initialize \quad \theta_0 = [w_0, b_0]; \\
while \quad t++ < max_{iterations} \text{ do} \\
| \quad \theta_{t+1} \leftarrow \theta_t - \eta \nabla \theta_t; \\
end
\]
Recall our gradient descent algorithm

We can write it more concisely as

**Algorithm:** gradient\_descent()

\[
t ← 0;
max\_iterations ← 1000;
\]

*Initialize* \( \theta_0 = [w_0, b_0] \);

*while* \( t++ < max\_iterations \) *do*

\[
\theta_{t+1} ← \theta_t - \eta \nabla \theta_t;
\]

*end*

where \( \nabla \theta_t = \left[ \frac{\partial \mathcal{L}(\theta)}{\partial w_t}, \frac{\partial \mathcal{L}(\theta)}{\partial b_t} \right]^T \)
Recall our gradient descent algorithm.

We can write it more concisely as

**Algorithm: gradient_descent()**

\[
t ← 0;
\]
\[
max\_iterations ← 1000;
\]
\[
Initialize \ θ_0 = [w_0, b_0];
\]
\[
while \ t++ < max\_iterations \ do

| \( \theta_{t+1} ← \theta_t - \eta \nabla \theta_t; \)

end
\]

where \( \nabla \theta_t = \left[ \frac{\partial L(\theta)}{\partial w_t}, \frac{\partial L(\theta)}{\partial b_t} \right]^T \)

Now, in this feedforward neural network, instead of \( \theta = [w, b] \) we have \( \theta = [W_1, W_2, \ldots, W_L, b_1, b_2, \ldots, b_L] \)
Recall our gradient descent algorithm

We can write it more concisely as

**Algorithm: gradient_descent()**

\[
t ← 0;
max_{iterations} ← 1000;
Initialize \ \theta_0 = [w_0, b_0];
\]

[While loop]

\[
θ_{t+1} ← θ_t − η \nabla_θ t;
\]

[End loop]

where \ \nabla_θ t = \[\frac{∂L(θ)}{∂w_t}, \frac{∂L(θ)}{∂b_t}\]^T

Now, in this feedforward neural network, instead of \ \theta = [w, b] we have \ \theta = [W_1, W_2, .., W_L, b_1, b_2, .., b_L]

We can still use the same algorithm for learning the parameters of our model
Recall our gradient descent algorithm

We can write it more concisely as

**Algorithm:** gradient_descent()

\[
t \leftarrow 0;
\]
\[
\text{max}_\text{iterations} \leftarrow 1000;
\]
\[
\text{Initialize } \theta_0 = [W_0^0, ..., W_L^0, b_0^0, ..., b_L^0];
\]
\[
\text{while } t++ < \text{max}_\text{iterations} \text{ do}
\]
\[
\theta_{t+1} \leftarrow \theta_t - \eta \nabla \theta_t;
\]
\[
\text{end}
\]

where \( \nabla \theta_t = [\frac{\partial \mathcal{L}(\theta)}{\partial W_{1,t}}, ..., \frac{\partial \mathcal{L}(\theta)}{\partial W_{L,t}}, \frac{\partial \mathcal{L}(\theta)}{\partial b_{1,t}}, ..., \frac{\partial \mathcal{L}(\theta)}{\partial b_{L,t}}]^T \)

Now, in this feedforward neural network, instead of \( \theta = [w, b] \) we have \( \theta = [W_1, W_2, ..., W_L, b_1, b_2, ..., b_L] \)

We can still use the same algorithm for learning the parameters of our model
Except that now our $\nabla \theta$ looks much more nasty
Except that now our $\nabla \theta$ looks much more nasty

$$\begin{bmatrix}
\frac{\partial L(\theta)}{\partial W_{111}} \\
\vdots \\
\frac{\partial L(\theta)}{\partial W_{1n}} \\
\vdots \\
\frac{\partial L(\theta)}{\partial W_{L,1}} \\
\vdots \\
\frac{\partial L(\theta)}{\partial W_{L,k}} \\
\vdots \\
\frac{\partial L(\theta)}{\partial b_{11}} \\
\vdots \\
\frac{\partial L(\theta)}{\partial b_{L,1}} \\
\vdots \\
\frac{\partial L(\theta)}{\partial b_{L,k}} \\
\vdots
\end{bmatrix}$$
Except that now our $\nabla \theta$ looks much more nasty

$$\begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} & \cdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} & \cdots \\
\vdots & \ddots & \ddots
\end{bmatrix}$$
Except that now our $\nabla \theta$ looks much more nasty

$$
\begin{bmatrix}
\frac{\partial L(\theta)}{\partial W_{111}} & \cdots & \frac{\partial L(\theta)}{\partial W_{11n}} \\
\vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial W_{L,11}} & \cdots & \frac{\partial L(\theta)}{\partial W_{L,n}} \\
\frac{\partial L(\theta)}{\partial b_{11}} & \cdots & \frac{\partial L(\theta)}{\partial b_{L,n}} \\
\vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial W_{1nn}} & \cdots & \frac{\partial L(\theta)}{\partial W_{nn}} \\
\end{bmatrix}
$$
Except that now our $\nabla \theta$ looks much more nasty

$$\nabla \theta$$

is thus composed of

$$\nabla W_{11}, \nabla W_{21}, \ldots, \nabla W_{L-1n}, \nabla W_L \in \mathbb{R}^{n \times n},$$

$$\nabla b_1, \nabla b_2, \ldots, \nabla b_{L-1} \in \mathbb{R}^{n}$$

and

$$\nabla b_L \in \mathbb{R}^{k}.$$
Except that now our $\nabla \theta$ looks much more nasty

$$
\begin{bmatrix}
\frac{\partial L(\theta)}{\partial W_{111}} & \cdots & \frac{\partial L(\theta)}{\partial W_{11}} & \frac{\partial L(\theta)}{\partial W_{11}} & \cdots & \frac{\partial L(\theta)}{\partial W_{21n}} \\
\frac{\partial L(\theta)}{\partial W_{121}} & \cdots & \frac{\partial L(\theta)}{\partial W_{12}} & \frac{\partial L(\theta)}{\partial W_{12}} & \cdots & \frac{\partial L(\theta)}{\partial W_{22n}} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial W_{1n1}} & \cdots & \frac{\partial L(\theta)}{\partial W_{1n}} & \frac{\partial L(\theta)}{\partial W_{1n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{2nn}}
\end{bmatrix}
$$
Except that now our $\nabla \theta$ looks much more nasty

$$
\frac{\partial L(\theta)}{\partial W_{111}} \cdots \frac{\partial L(\theta)}{\partial W_{11n}} \frac{\partial L(\theta)}{\partial W_{121}} \cdots \frac{\partial L(\theta)}{\partial W_{12n}} \cdots \frac{\partial L(\theta)}{\partial W_{1n1}} \\
\frac{\partial L(\theta)}{\partial W_{11n}} \frac{\partial L(\theta)}{\partial W_{12n}} \frac{\partial L(\theta)}{\partial W_{211}} \cdots \frac{\partial L(\theta)}{\partial W_{21n}} \cdots \frac{\partial L(\theta)}{\partial W_{21n}} \\
\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
\frac{\partial L(\theta)}{\partial W_{1nn}} \frac{\partial L(\theta)}{\partial W_{2nn}} \cdots \frac{\partial L(\theta)}{\partial W_{2nn}} \cdots
$$
Except that now our $\nabla \theta$ looks much more nasty

\[
\begin{bmatrix}
\frac{\partial L(\theta)}{\partial W_{11}} & \cdots & \frac{\partial L(\theta)}{\partial W_{1n}} & \frac{\partial L(\theta)}{\partial W_{21}} & \cdots & \frac{\partial L(\theta)}{\partial W_{2n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{L,1}} & \frac{\partial L(\theta)}{\partial W_{L,1k}} \\
\frac{\partial L(\theta)}{\partial W_{12}} & \cdots & \frac{\partial L(\theta)}{\partial W_{1n}} & \frac{\partial L(\theta)}{\partial W_{22}} & \cdots & \frac{\partial L(\theta)}{\partial W_{2n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{L,21}} & \frac{\partial L(\theta)}{\partial W_{L,2k}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial L(\theta)}{\partial W_{1n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{1n}} & \frac{\partial L(\theta)}{\partial W_{2n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{2n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{L,n}} & \frac{\partial L(\theta)}{\partial W_{L,nk}}
\end{bmatrix}
\]
Except that now our $\nabla \theta$ looks much more nasty

$$\nabla \theta = \begin{bmatrix}
\frac{\partial L(\theta)}{\partial W_{111}} & \cdots & \frac{\partial L(\theta)}{\partial W_{11n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{1n1}} \\
\frac{\partial L(\theta)}{\partial W_{121}} & \cdots & \frac{\partial L(\theta)}{\partial W_{12n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{1nn}} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial W_{L11}} & \cdots & \frac{\partial L(\theta)}{\partial W_{L1k}} & \cdots & \frac{\partial L(\theta)}{\partial b_{11}} \\
\frac{\partial L(\theta)}{\partial W_{L21}} & \cdots & \frac{\partial L(\theta)}{\partial W_{L2k}} & \cdots & \frac{\partial L(\theta)}{\partial b_{12}} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial W_{L1n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{L1k}} & \cdots & \frac{\partial L(\theta)}{\partial b_{L1}} \\
\frac{\partial L(\theta)}{\partial W_{L2n}} & \cdots & \frac{\partial L(\theta)}{\partial W_{L2k}} & \cdots & \frac{\partial L(\theta)}{\partial b_{L2}} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial b_{11}} & \cdots & \frac{\partial L(\theta)}{\partial b_{12}} & \cdots & \frac{\partial L(\theta)}{\partial b_{L1}} \\
\frac{\partial L(\theta)}{\partial b_{12}} & \cdots & \frac{\partial L(\theta)}{\partial b_{1k}} & \cdots & \frac{\partial L(\theta)}{\partial b_{L2}} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial b_{L1}} & \cdots & \frac{\partial L(\theta)}{\partial b_{Lk}} & \cdots & \frac{\partial L(\theta)}{\partial b_{Lk}}
\end{bmatrix}$$
Except that now our $\nabla \theta$ looks much more nasty

$$
\begin{bmatrix}
\frac{\partial L(\theta)}{\partial W_{11}} & \frac{\partial L(\theta)}{\partial W_{1n}} & \frac{\partial L(\theta)}{\partial W_{21}} & \ldots & \frac{\partial L(\theta)}{\partial W_{1n}} & \frac{\partial L(\theta)}{\partial W_{L,1}} & \frac{\partial L(\theta)}{\partial W_{L,1k}} & \frac{\partial L(\theta)}{\partial b_{11}} & \ldots & \frac{\partial L(\theta)}{\partial b_{L1}} \\
\frac{\partial L(\theta)}{\partial W_{12}} & \frac{\partial L(\theta)}{\partial W_{12n}} & \frac{\partial L(\theta)}{\partial W_{22}} & \ldots & \frac{\partial L(\theta)}{\partial W_{12n}} & \frac{\partial L(\theta)}{\partial W_{L,2}} & \frac{\partial L(\theta)}{\partial W_{L,2k}} & \frac{\partial L(\theta)}{\partial b_{12}} & \ldots & \frac{\partial L(\theta)}{\partial b_{L2}} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial W_{1n1}} & \frac{\partial L(\theta)}{\partial W_{1nn}} & \frac{\partial L(\theta)}{\partial W_{2n1}} & \ldots & \frac{\partial L(\theta)}{\partial W_{1nn}} & \frac{\partial L(\theta)}{\partial W_{L,n}} & \frac{\partial L(\theta)}{\partial W_{L,nk}} & \frac{\partial L(\theta)}{\partial b_{1n}} & \ldots & \frac{\partial L(\theta)}{\partial b_{Lk}}
\end{bmatrix}
$$

$\nabla \theta$ is thus composed of
$$\nabla W_1, \nabla W_2, \ldots \nabla W_{L-1} \in \mathbb{R}^{n \times n}, \nabla W_L \in \mathbb{R}^{n \times k}, \nabla b_1, \nabla b_2, \ldots, \nabla b_{L-1} \in \mathbb{R}^n \text{ and } \nabla b_L \in \mathbb{R}^k$$
We need to answer two questions

1. How to choose the loss function $L(\theta)$?
2. How to compute $\nabla \theta$ which is composed of $\nabla W_1, \nabla W_2, \ldots, \nabla W_{L-1} \in \mathbb{R}^{n \times n}, \nabla W_L \in \mathbb{R}^{n \times k}$, $\nabla b_1, \nabla b_2, \ldots, \nabla b_{L-1} \in \mathbb{R}^{n}$ and $\nabla b_L \in \mathbb{R}^{k}$?
We need to answer two questions

- How to choose the loss function $L(\theta)$?
We need to answer two questions

- How to choose the loss function $\mathcal{L}(\theta)$?
- How to compute $\nabla \theta$ which is composed of $\nabla W_1, \nabla W_2, ..., \nabla W_{L-1} \in \mathbb{R}^{n \times n}, \nabla W_L \in \mathbb{R}^{n \times k}$, $\nabla b_1, \nabla b_2, ..., \nabla b_{L-1} \in \mathbb{R}^n$ and $\nabla b_L \in \mathbb{R}^k$?
Module 4.3: Output Functions and Loss Functions
We need to answer two questions

- How to choose the loss function $\mathcal{L}(\theta)$?
- How to compute $\nabla \theta$ which is composed of:
  - $\nabla W_1, \nabla W_2, ..., \nabla W_{L-1} \in \mathbb{R}^{n \times n}, \nabla W_L \in \mathbb{R}^{n \times k}$
  - $\nabla b_1, \nabla b_2, ..., \nabla b_{L-1} \in \mathbb{R}^n$ and $\nabla b_L \in \mathbb{R}^k$?
We need to answer two questions

- How to choose the loss function $L(\theta)$?
- How to compute $\nabla \theta$ which is composed of:
  $\nabla W_1, \nabla W_2, \ldots, \nabla W_{L-1} \in \mathbb{R}^{n \times n}$, $\nabla W_L \in \mathbb{R}^{n \times k}$
  $\nabla b_1, \nabla b_2, \ldots, \nabla b_{L-1} \in \mathbb{R}^n$ and $\nabla b_L \in \mathbb{R}^k$?
The choice of loss function depends on the problem at hand.

The loss function should capture how much $\hat{y}_i$ deviates from $y_i$.

If $y_i \in \mathbb{R}^n$, then the squared error loss can capture this deviation:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{3} (\hat{y}_{ij} - y_{ij})^2$$
The choice of loss function depends on the problem at hand.

We will illustrate this with the help of two examples.

\[ y_i = \{ 7.5, 8.2, 7.7 \} \]

The choice of loss function depends on the problem at hand.

We will illustrate this with the help of two examples.
The choice of loss function depends on the problem at hand.

We will illustrate this with the help of two examples.

Consider our movie example again but this time we are interested in predicting ratings.

\[ y_i = \{7.5, 8.2, 7.7\} \]

- **imdb Rating**
- **Critic Rating**
- **RT Rating**

The loss function should capture how \( \hat{y}_i \) deviates from \( y_i \).

If \( y_i \in \mathbb{R}^n \) then the squared error loss can capture this deviation.

\[
L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{3} (\hat{y}_{ij} - y_{ij})^2
\]
The choice of loss function depends on the problem at hand.

We will illustrate this with the help of two examples.

Consider our movie example again but this time we are interested in predicting ratings.

Here $y_i \in \mathbb{R}^3$
The choice of loss function depends on the problem at hand.

We will illustrate this with the help of two examples.

Consider our movie example again but this time we are interested in predicting ratings.

Here $y_i \in \mathbb{R}^3$.

The loss function should capture how much $\hat{y}_i$ deviates from $y_i$.
\[ y_i = \{7.5 \quad 8.2 \quad 7.7\} \]

<table>
<thead>
<tr>
<th>imdb</th>
<th>Critics</th>
<th>RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>Rating</td>
<td>Rating</td>
</tr>
</tbody>
</table>

The choice of loss function depends on the problem at hand.

- We will illustrate this with the help of two examples.
- Consider our movie example again but this time we are interested in predicting ratings.
- Here \( y_i \in \mathbb{R}^3 \).
- The loss function should capture how much \( \hat{y}_i \) deviates from \( y_i \).
- If \( y_i \in \mathbb{R}^n \) then the squared error loss can capture this deviation.

\[
\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{3} (\hat{y}_{ij} - y_{ij})^2
\]
A related question: What should the output function ‘O’ be if $y_i \in \mathbb{R}$?

$h_L = \hat{y} = f(x)$
A related question: What should the output function ‘\(O\)’ be if \(y_i \in \mathbb{R}\)?

More specifically, can it be the logistic function?
A related question: What should the output function ‘O’ be if $y_i \in \mathbb{R}$?

More specifically, can it be the logistic function?

No, because it restricts $\hat{y}_i$ to a value between 0 & 1 but we want $\hat{y}_i \in \mathbb{R}$.
A related question: What should the output function ‘O’ be if \( y_i \in \mathbb{R} \)?

More specifically, can it be the logistic function?

No, because it restricts \( \hat{y}_i \) to a value between 0 & 1 but we want \( \hat{y}_i \in \mathbb{R} \)

So, in such cases it makes sense to have ‘O’ as linear function

\[
\hat{y}_i = f(x_i) = h_L = O(a_L) = W_O a_L + b_O
\]
A related question: What should the output function ‘\(O\)' be if \(y_i \in \mathbb{R}\)?

More specifically, can it be the logistic function?

No, because it restricts \(\hat{y}_i\) to a value between 0 & 1 but we want \(\hat{y}_i \in \mathbb{R}\).

So, in such cases it makes sense to have ‘\(O\)' as linear function

\[ f(x) = h_L = O(a_L) = W_Oa_L + b_O \]

\(\hat{y}_i = f(x_i)\) is no longer bounded between 0 and 1.
Now let us consider another problem for which a different loss function would be appropriate.

\[ y = [1 \ 0 \ 0 \ 0] \]

Apple  Mango  Orange  Banana

Neural network with \( L - 1 \) hidden layers
Now let us consider another problem for which a different loss function would be appropriate.

Suppose we want to classify an image into 1 of $k$ classes.
Now let us consider another problem for which a different loss function would be appropriate.

Suppose we want to classify an image into 1 of $k$ classes.

Here again we could use the squared error loss to capture the deviation.
Now let us consider another problem for which a different loss function would be appropriate.

Suppose we want to classify an image into 1 of \( k \) classes.

Here again we could use the squared error loss to capture the deviation.

But can you think of a better function?
Neural network with $L - 1$ hidden layers

$y = [1 \ 0 \ 0 \ 0]$  
Apple  Mango  Orange  Banana

Notice that $y$ is a probability distribution

What choice of the output activation $\hat{y}$ will ensure this?

$$a_L = W_L a_{L-1} + b_L$$

$$\hat{y}_j = O(a_L)_j = e^{a_{L,j}} \sum_{i=1} e^{a_{L,i}}$$

$\hat{y}_j$ is the $j$th element of $\hat{y}$ and $a_{L,j}$ is the $j$th element of the vector $a_L$. This function is called the softmax function.
Neural network with $L - 1$ hidden layers

$y = [1 \ 0 \ 0 \ 0]$  
Apple   Mango   Orange   Banana

Notice that $y$ is a probability distribution
Therefore we should also ensure that $\hat{y}$ is a probability distribution
Notice that $y$ is a probability distribution

Therefore we should also ensure that $\hat{y}$ is a probability distribution

What choice of the output activation ‘$O$’ will ensure this?

$$a_L = W_L h_{L-1} + b_L$$
Notice that \( y \) is a probability distribution

Therefore we should also ensure that \( \hat{y} \) is a probability distribution

What choice of the output activation ‘\( O \)’ will ensure this?

\[
a_L = W_L h_{L-1} + b_L
\]

\[
\hat{y}_j = O(a_L)_j = \frac{e^{a_{L,j}}}{\sum_{i=1}^k e^{a_{L,i}}}
\]

\( O(a_L)_j \) is the \( j^{th} \) element of \( \hat{y} \) and \( a_{L,j} \) is the \( j^{th} \) element of the vector \( a_L \).
Notice that $y$ is a probability distribution.

Therefore we should also ensure that $\hat{y}$ is a probability distribution.

What choice of the output activation ‘$O$’ will ensure this?

$$a_L = W_L h_{L-1} + b_L$$

$$\hat{y}_j = O(a_L)_j = \frac{e^{a_{L,j}}}{\sum_{i=1}^k e^{a_{L,i}}}$$

$O(a_L)_j$ is the $j^{th}$ element of $\hat{y}$ and $a_{L,j}$ is the $j^{th}$ element of the vector $a_L$.

This function is called the softmax function.
$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$

Apple    Mango    Orange    Banana

Now that we have ensured that both $y$ & $\hat{y}$ are probability distributions can you think of a function which captures the difference between them?

Cross-entropy

$L(\theta) = -\sum_{c=1}^{k} y_c \log \hat{y}_c$

Notice that $y_c = 1$ if $c = \ell$ (the true class label) $= 0$ otherwise

∴ $L(\theta) = -\log \hat{y}_\ell$
Now that we have ensured that both $y$ & $\hat{y}$ are probability distributions can you think of a function which captures the difference between them?

- **Cross-entropy**

$$
\mathcal{L}(\theta) = - \sum_{c=1}^{k} y_c \log \hat{y}_c
$$
Now that we have ensured that both $y$ & $\hat{y}$ are probability distributions, can you think of a function which captures the difference between them?

- **Cross-entropy**

  $$\mathcal{L}(\theta) = -\sum_{c=1}^{k} y_c \log \hat{y}_c$$

- Notice that

  $$y_c = 1 \text{ if } c = \ell \text{ (the true class label)}$$
  $$= 0 \text{ otherwise}$$

  $\therefore \mathcal{L}(\theta) = -\log \hat{y}_\ell$
So, for classification problem (where you have to choose 1 of $K$ classes), we use the following objective function

$$
\min_{\theta} \mathcal{L}(\theta) = -\log \hat{y}_\ell
$$

or

$$
\max_{\theta} -\mathcal{L}(\theta) = \log \hat{y}_\ell
$$
So, for classification problem (where you have to choose 1 of $K$ classes), we use the following objective function

$$
\text{minimize} \quad \mathcal{L}(\theta) = -\log \hat{y}_\ell \\
\text{or maximize} \quad -\mathcal{L}(\theta) = \log \hat{y}_\ell
$$

But wait!
Is $\hat{y}_\ell$ a function of $\theta = [W_1, W_2, \ldots, W_L, b_1, b_2, \ldots, b_L]$?
\[ h_L = \hat{y} = f(x) \]

- So, for classification problem (where you have to choose 1 of \( K \) classes), we use the following objective function

\[
\begin{align*}
\text{minimize} & \quad \mathcal{L}(\theta) = - \log \hat{y}_\ell \\
\text{or maximize} & \quad - \mathcal{L}(\theta) = \log \hat{y}_\ell
\end{align*}
\]

- But wait! Is \( \hat{y}_\ell \) a function of \( \theta = [W_1, W_2, \ldots, W_L, b_1, b_2, \ldots, b_L] \)?
- Yes, it is indeed a function of \( \theta \)

\[
\hat{y}_\ell = [O(W_3g(W_2g(W_1x + b_1) + b_2) + b_3)]_\ell
\]

What does \( \hat{y}_\ell \) encode?

It is the probability that \( x \) belongs to the \( \ell \)th class (bring it as close to 1). \( \log \hat{y}_\ell \) is called the log-likelihood of the data.
\[ h_L = \hat{y} = f(x) \]

- So, for classification problem (where you have to choose 1 of \( K \) classes), we use the following objective function

\[
\min_\theta \mathcal{L}(\theta) = -\log \hat{y}_\ell \\
\text{or maximize } \max_\theta -\mathcal{L}(\theta) = \log \hat{y}_\ell
\]

- But wait!
  Is \( \hat{y}_\ell \) a function of \( \theta = [W_1, W_2, .., W_L, b_1, b_2, .., b_L] \)?
- Yes, it is indeed a function of \( \theta \)

\[
\hat{y}_\ell = [O(W_3g(W_2g(W_1x + b_1) + b_2) + b_3)]_\ell
\]

- What does \( \hat{y}_\ell \) encode?

It is the probability that \( x \) belongs to the \( \ell \)th class (bring it as close to 1).

\( \log \hat{y}_\ell \) is called the log-likelihood of the data.
\[ h_L = \hat{y} = f(x) \]

- So, for classification problem (where you have to choose 1 of \( K \) classes), we use the following objective function
  
  \[
  \underset{\theta}{\text{minimize}} \quad \mathcal{L}(\theta) = -\log \hat{y}_\ell \\
  \text{or} \quad \underset{\theta}{\text{maximize}} \quad -\mathcal{L}(\theta) = \log \hat{y}_\ell
  \]

- But wait!
  
  Is \( \hat{y}_\ell \) a function of \( \theta = [W_1, W_2, \ldots, W_L, b_1, b_2, \ldots, b_L] \)?

- Yes, it is indeed a function of \( \theta \)
  
  \[
  \hat{y}_\ell = [O(W_3 g(W_2 g(W_1 x + b_1) + b_2) + b_3)]_\ell
  \]

- What does \( \hat{y}_\ell \) encode?
  
  It is the probability that \( x \) belongs to the \( \ell^{th} \) class (bring it as close to 1).
• So, for classification problem (where you have to choose 1 of $K$ classes), we use the following objective function

$$
\minimize_{\theta} \mathcal{L}(\theta) = -\log \hat{y}_\ell \\
\text{or} \quad \maximize_{\theta} -\mathcal{L}(\theta) = \log \hat{y}_\ell
$$

• But wait!
  Is $\hat{y}_\ell$ a function of $\theta = [W_1, W_2, .., W_L, b_1, b_2, .., b_L]$?

• Yes, it is indeed a function of $\theta$

$$
\hat{y}_\ell = [O(W_3g(W_2g(W_1x + b_1) + b_2) + b_3)]_\ell
$$

• What does $\hat{y}_\ell$ encode?

• It is the probability that $x$ belongs to the $\ell^{th}$ class (bring it as close to 1).

• $\log \hat{y}_\ell$ is called the log-likelihood of the data.
<table>
<thead>
<tr>
<th>Outputs</th>
<th>Real Values</th>
<th>Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Activation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loss Function</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Of course, there could be other loss functions depending on the problem at hand, but the two loss functions that we just saw are encountered very often. For the rest of this lecture, we will focus on the case where the output activation is a softmax function and the loss function is cross entropy.
<table>
<thead>
<tr>
<th>Outputs</th>
<th>Real Values</th>
<th>Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Activation</td>
<td>Linear</td>
<td></td>
</tr>
<tr>
<td>Loss Function</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Of course, there could be other loss functions depending on the problem at hand but the two loss functions that we just saw are encountered very often. For the rest of this lecture we will focus on the case where the output activation is a softmax function and the loss function is cross entropy.
<table>
<thead>
<tr>
<th>Output Activation</th>
<th>Real Values</th>
<th>Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td></td>
<td>Softmax</td>
</tr>
</tbody>
</table>

Loss Function

Of course, there could be other loss functions depending on the problem at hand but the two loss functions that we just saw are encountered very often. For the rest of this lecture we will focus on the case where the output activation is a softmax function and the loss function is cross entropy.
### Outputs

<table>
<thead>
<tr>
<th>Output Activation</th>
<th>Real Values</th>
<th>Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td></td>
<td>Softmax</td>
</tr>
<tr>
<td>Loss Function</td>
<td>Squared Error</td>
<td></td>
</tr>
<tr>
<td>Outputs</td>
<td>Real Values</td>
<td>Probabilities</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Output Activation</td>
<td>Linear</td>
<td>Softmax</td>
</tr>
<tr>
<td>Loss Function</td>
<td>Squared Error</td>
<td>Cross Entropy</td>
</tr>
</tbody>
</table>
Of course, there could be other loss functions depending on the problem at hand but the two loss functions that we just saw are encountered very often.
Of course, there could be other loss functions depending on the problem at hand but the two loss functions that we just saw are encountered very often.

For the rest of this lecture we will focus on the case where the output activation is a softmax function and the loss function is cross entropy.
Module 4.4: Backpropagation (Intuition)
We need to answer two questions

- How to choose the loss function $\mathcal{L}(\theta)$?
- How to compute $\nabla \theta$ which is composed of:
  - $\nabla W_1, \nabla W_2, ..., \nabla W_{L-1} \in \mathbb{R}^{n \times n}, \nabla W_L \in \mathbb{R}^{n \times k}$
  - $\nabla b_1, \nabla b_2, ..., \nabla b_{L-1} \in \mathbb{R}^{n}$ and $\nabla b_L \in \mathbb{R}^{k}$?
We need to answer two questions

- How to choose the loss function $\mathcal{L}(\theta)$?
- How to compute $\nabla \theta$ which is composed of:
  - $\nabla W_1, \nabla W_2, \ldots, \nabla W_{L-1} \in \mathbb{R}^{n \times n}$, $\nabla W_L \in \mathbb{R}^{n \times k}$
  - $\nabla b_1, \nabla b_2, \ldots, \nabla b_{L-1} \in \mathbb{R}^n$ and $\nabla b_L \in \mathbb{R}^k$?
Let us focus on this one weight \((W_{112})\).

\[ \hat{y} = f(x) \]

**Algorithm:** gradient descent()

\[
t \leftarrow 0; \\
max\_iterations \leftarrow 1000; \\
Initialize \theta_0; \\
while \ t++ < max\_iterations \\
do \\
\[ \theta_{t+1} \leftarrow \theta_t - \eta \nabla \theta_t; \] \\
end
Let us focus on this one weight ($W_{112}$).

To learn this weight using SGD we need a formula for $\frac{\partial L(\theta)}{\partial W_{112}}$.

Algorithm: gradient descent()

$t \leftarrow 0$;
$max\_iterations \leftarrow 1000$;
Initialize $\theta_0$;
while $t++ < max\_iterations$
do
  $\theta_{t+1} \leftarrow \theta_t - \eta \nabla \theta_t$;
end
Let us focus on this one weight \( W_{112} \).

To learn this weight using SGD we need a formula for \( \frac{\partial L(\theta)}{\partial W_{112}} \).

We will see how to calculate this.

\[
\hat{y} = f(x)
\]

\[
\begin{align*}
\text{Algorithm:} & \quad \text{gradient descent}() \\
& \quad t \leftarrow 0; \\
& \quad max\_iterations \leftarrow 1000; \\
& \quad \text{Initialize } \theta_0; \\
& \quad \text{while } t++ < max\_iterations \\
& \quad \quad \text{do} \\
& \quad \quad \quad \theta_{t+1} \leftarrow \theta_t - \eta \nabla \theta_t; \\
& \quad \quad \text{end}
\end{align*}
\]
First let us take the simple case when we have a deep but thin network.
• First let us take the simple case when we have a deep but thin network.
• In this case it is easy to find the derivative by chain rule.
First let us take the simple case when we have a deep but thin network. In this case it is easy to find the derivative by chain rule.

$$\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} = \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{L11}} \frac{\partial a_{L11}}{\partial h_{21}} \frac{\partial h_{21}}{\partial a_{21}} \frac{\partial a_{21}}{\partial h_{11}} \frac{\partial h_{11}}{\partial a_{11}} \frac{\partial a_{11}}{\partial W_{111}}$$
First let us take the simple case when we have a deep but thin network.

In this case it is easy to find the derivative by chain rule.

\[
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} = \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{L11}} \frac{\partial a_{L11}}{\partial h_{21}} \frac{\partial h_{21}}{\partial a_{21}} \frac{\partial a_{21}}{\partial h_{11}} \frac{\partial h_{11}}{\partial a_{11}} \frac{\partial a_{11}}{\partial W_{111}}
\]

\[
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} = \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}} \frac{\partial a_{L11}}{\partial h_{11}} \frac{\partial h_{11}}{\partial W_{111}}
\]

(just compressing the chain rule)
First let us take the simple case when we have a deep but thin network.

In this case it is easy to find the derivative by chain rule.

\[
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} = \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{L11}} \frac{\partial a_{L11}}{\partial h_{21}} \frac{\partial h_{21}}{\partial a_{21}} \frac{\partial a_{21}}{\partial h_{11}} \frac{\partial h_{11}}{\partial a_{11}} \frac{\partial a_{11}}{\partial W_{111}}
\]

\[
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{11}} \frac{\partial h_{11}}{\partial W_{111}}
\]

(just compressing the chain rule)

\[
\frac{\partial \mathcal{L}(\theta)}{\partial W_{211}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{21}} \frac{\partial h_{21}}{\partial W_{211}}
\]
First let us take the simple case when we have a deep but thin network.

In this case it is easy to find the derivative by chain rule.

\[
\begin{align*}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} &= \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{L1}} \frac{\partial a_{L1}}{\partial h_{21}} \frac{\partial h_{21}}{\partial a_{21}} \frac{\partial a_{21}}{\partial h_{11}} \frac{\partial h_{11}}{\partial a_{11}} \frac{\partial a_{11}}{\partial W_{111}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} &= \frac{\partial \mathcal{L}(\theta)}{\partial h_{11}} \frac{\partial h_{11}}{\partial W_{111}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{211}} &= \frac{\partial \mathcal{L}(\theta)}{\partial h_{21}} \frac{\partial h_{21}}{\partial W_{211}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{L11}} &= \frac{\partial \mathcal{L}(\theta)}{\partial a_{L1}} \frac{\partial a_{L1}}{\partial W_{L11}}
\end{align*}
\]
Let us see an intuitive explanation of backpropagation before we get into the mathematical details.
We get a certain loss at the output and we try to figure out who is responsible for this loss.
• We get a certain loss at the output and we try to figure out who is responsible for this loss

• So, we talk to the output layer and say “Hey! You are not producing the desired output, better take responsibility”.

\[
-f(x) = \hat{y}_\ell = O(W_L h_{L-1} + b_L)
\]
• We get a certain loss at the output and we try to figure out who is responsible for this loss.

• So, we talk to the output layer and say “Hey! You are not producing the desired output, better take responsibility”.

• The output layer says “Well, I take responsibility for my part but please understand that I am only as good as the hidden layer and weights below me”. After all …

$$f(x) = \hat{y} = O(W_L h_{L-1} + b_L)$$
So, we talk to $W_L, b_L$ and $h_L$ and ask them “What is wrong with you?”
So, we talk to \( W_L, b_L \) and \( h_L \) and ask them “What is wrong with you?”

\( W_L \) and \( b_L \) take full responsibility but \( h_L \) says “Well, please understand that I am only as good as the pre-activation layer”

The pre-activation layer in turn says that I am only as good as the hidden layer and weights below me. We continue in this manner and realize that the responsibility lies with all the weights and biases (i.e. all the parameters of the model)

But instead of talking to them directly, it is easier to talk to them through the hidden layers and output layers (and this is exactly what the chain rule allows us to do)

\[
\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial \hat{y}_\ell} \frac{\partial \hat{y}_\ell}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_1} \frac{\partial W_1}{\partial x_n}
\]

"Talking to the output layer"

"Talking to the previous hidden layer"

"Talking to the previous hidden layer"

"Talking to the weights"
So, we talk to $W_L, b_L$ and $h_L$ and ask them “What is wrong with you?”

$W_L$ and $b_L$ take full responsibility but $h_L$ says “Well, please understand that I am only as good as the pre-activation layer”

The pre-activation layer in turn says that I am only as good as the hidden layer and weights below me.
So, we talk to $W_L, b_L$ and $h_L$ and ask them “What is wrong with you?”

$W_L$ and $b_L$ take full responsibility but $h_L$ says “Well, please understand that I am only as good as the pre-activation layer”

The pre-activation layer in turn says that I am only as good as the hidden layer and weights below me.

We continue in this manner and realize that the responsibility lies with all the weights and biases (i.e. all the parameters of the model)

![Neural network diagram](image-url)

$\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ \hspace{1cm} $- \log \hat{y}_\ell$

$W_1, a_1$ \hspace{1cm} $W_2, a_2$ \hspace{1cm} $W_3, a_3$

$b_1, h_1$ \hspace{1cm} $b_2, h_2$ \hspace{1cm} $b_3$
So, we talk to $W_L, b_L$ and $h_L$ and ask them “What is wrong with you?”

$W_L$ and $b_L$ take full responsibility but $h_L$ says “Well, please understand that I am only as good as the pre-activation layer”

The pre-activation layer in turn says that I am only as good as the hidden layer and weights below me.

We continue in this manner and realize that the responsibility lies with all the weights and biases (i.e. all the parameters of the model)

But instead of talking to them directly, it is easier to talk to them through the hidden layers and output layers (and this is exactly what the chain rule allows us to do)
So, we talk to $W_L, b_L$ and $h_L$ and ask them “What is wrong with you?”

$W_L$ and $b_L$ take full responsibility but $h_L$ says “Well, please understand that I am only as good as the pre-activation layer”

The pre-activation layer in turn says that I am only as good as the hidden layer and weights below me.

We continue in this manner and realize that the responsibility lies with all the weights and biases (i.e. all the parameters of the model).

But instead of talking to them directly, it is easier to talk to them through the hidden layers and output layers (and this is exactly what the chain rule allows us to do)

\[
\frac{\partial L(\theta)}{\partial W_{111}} = \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_{111}}
\]

Talk to the weight directly

Talk to the output layer

Talk to the previous hidden layer

Talk to the previous hidden layer

and now talk to the weights

- Mitesh M. Khapra
Quantities of interest (roadmap for the remaining part):

- Gradient w.r.t. output units
- Gradient w.r.t. hidden units
- Gradient w.r.t. weights and biases

\[
\frac{\partial L(\theta)}{\partial W_{111}} = \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_{111}}
\]

- Talk to the weight directly
- Talk to the output layer
- Talk to the previous hidden layer
- Talk to the previous hidden layer
- and now talk to the weights
Quantities of interest (roadmap for the remaining part):

\[
\frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} = \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_3} \cdot \frac{\partial a_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial a_2} \cdot \frac{\partial a_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial a_1} \cdot \frac{\partial a_1}{\partial W_{111}}
\]

- Talk to the weight directly
- Talk to the output layer
- Talk to the previous hidden layer
- Talk to the previous hidden layer
- Talk to the weights

Our focus is on Cross entropy loss and Softmax output.
Quantities of interest (roadmap for the remaining part):
- Gradient w.r.t. output units

\[
\frac{\partial L(\theta)}{\partial W_{111}} = \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_{111}}
\]

Talk to the weight directly
Talk to the output layer
Talk to the previous hidden layer
Talk to the previous hidden layer
and now talk to the weights
Quantities of interest (roadmap for the remaining part):

- Gradient w.r.t. output units
- Gradient w.r.t. hidden units

\[
\frac{\partial L(\theta)}{\partial W_{111}} = \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_{111}}
\]

- Talk to the weight directly
- Talk to the output layer
- Talk to the previous hidden layer
- Talk to the previous hidden layer
- and now talk to the weights

Our focus is on Cross entropy loss and Softmax output.
Quantities of interest (roadmap for the remaining part):

- Gradient w.r.t. output units
- Gradient w.r.t. hidden units
- Gradient w.r.t. weights and biases

\[
\begin{align*}
\frac{\partial L(\theta)}{\partial W_{111}} &= \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_{111}} \\
\text{Talk to the weight directly} & \quad \text{Talk to the output layer} & \quad \text{Talk to the previous hidden layer} & \quad \text{Talk to the previous hidden layer} & \quad \text{and now talk to the weights}
\end{align*}
\]
Quantities of interest (roadmap for the remaining part):

- Gradient w.r.t. output units
- Gradient w.r.t. hidden units
- Gradient w.r.t. weights and biases

\[
\frac{\partial L(\theta)}{\partial W_{111}} = \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_{111}}
\]

Talk to the weight directly Talk to the output layer Talk to the previous hidden layer Talk to the previous hidden layer and now talk to the weights

- Our focus is on Cross entropy loss and Softmax output.
Module 4.5: Backpropagation: Computing Gradients w.r.t. the Output Units
Quantities of interest (roadmap for the remaining part):

- Gradient w.r.t. output units
- Gradient w.r.t. hidden units
- Gradient w.r.t. weights

\[
\frac{\partial L(\theta)}{\partial W_{111}} = \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \hat{a}_3} \frac{\partial \hat{a}_3}{\partial h_2} \frac{\partial h_2}{\partial \hat{a}_2} \frac{\partial \hat{a}_2}{\partial h_1} \frac{\partial h_1}{\partial \hat{a}_1} \frac{\partial \hat{a}_1}{\partial W_{111}}
\]

Talk to the weight directly

Talk to the output layer

Talk to the previous hidden layer

Talk to the previous hidden layer

and now talk to the weights

- Our focus is on Cross entropy loss and Softmax output.
Let us first consider the partial derivative w.r.t. $i$-th output
Let us first consider the partial derivative w.r.t. $i$-th output

$$\mathcal{L}(\theta) = -\log \hat{y}_\ell \ (\ell = \text{true class label})$$
Let us first consider the partial derivative w.r.t. $i$-th output

$$\mathcal{L}(\theta) = -\log \hat{y}_\ell \quad (\ell = \text{true class label})$$

$$\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) =$$
Let us first consider the partial derivative w.r.t. \( i \)-th output

\[
\mathcal{L}(\theta) = -\log \hat{y}_\ell \quad (\ell = \text{true class label})
\]

\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = \frac{\partial}{\partial \hat{y}_i} (-\log \hat{y}_\ell)
\]

More compactly,

\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = -\frac{1}{\ell = \hat{y}_\ell} \quad \text{if} \quad i = \ell \quad \text{otherwise}
\]
Let us first consider the partial derivative w.r.t. $i$-th output

$$\mathcal{L}(\theta) = - \log \hat{y}_\ell \quad (\ell = \text{true class label})$$

$$\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = \frac{\partial}{\partial \hat{y}_i} (- \log \hat{y}_\ell)$$

$$= - \frac{1}{\hat{y}_\ell} \quad \text{if } i = \ell$$
Let us first consider the partial derivative w.r.t. $i$-th output

$$
\mathcal{L}(\theta) = - \log \hat{y}_\ell \quad (\ell = \text{true class label})
$$

$$
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = \frac{\partial}{\partial \hat{y}_i} (- \log \hat{y}_\ell)
$$

$$
= - \frac{1}{\hat{y}_\ell} \quad \text{if } i = \ell
$$

$$
= 0 \quad \text{otherwise}
$$
Let us first consider the partial derivative w.r.t. $i$-th output

$$L(\theta) = -\log \hat{y}_\ell \ (\ell = \text{true class label})$$

$$\frac{\partial}{\partial \hat{y}_i} (L(\theta)) = \frac{\partial}{\partial \hat{y}_i} (-\log \hat{y}_\ell)$$

$$= -\frac{1}{\hat{y}_\ell} \quad \text{if} \quad i = \ell$$

$$= 0 \quad \text{otherwise}$$

More compactly,
Let us first consider the partial derivative w.r.t. \( i \)-th output

\[
\mathcal{L}(\theta) = -\log \hat{y}_\ell \quad (\ell = \text{true class label})
\]

\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = \frac{\partial}{\partial \hat{y}_i} (-\log \hat{y}_\ell)
\]

\[
= - \frac{1}{\hat{y}_\ell} \quad \text{if } i = \ell
\]

\[
= 0 \quad \text{otherwise}
\]

More compactly,

\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = - \frac{1_{(i=\ell)}}{\hat{y}_\ell}
\]
\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L} (\theta)) = - \frac{1_{(\ell=i)}}{\hat{y}_\ell}
\]
\[
\frac{\partial}{\partial \hat{y}_i} \left( \mathcal{L}(\theta) \right) = -\frac{1}{\hat{y}_\ell}
\]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)
We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\begin{align*}
\nabla_{\hat{y}} \mathcal{L}(\theta) &= \left[ \begin{array}{c}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_2} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k}
\end{array} \right] \\
&= - \frac{1}{\hat{y}_\ell} \mathbf{e}_\ell
\end{align*}
\]

where \( \mathbf{e}(\ell) \) is a \( k \)-dimensional vector whose \( \ell \)-th element is 1 and all other elements are 0.
\[
\frac{\partial}{\partial \hat{y}_i}(\mathcal{L}(\theta)) = -\frac{1(\ell=i)}{\hat{y}_\ell}
\]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1}
\vdots
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k}
\end{bmatrix}
\]

where \( e(\ell) \) is a \( k \)-dimensional vector whose \( \ell \)-th element is 1 and all other elements are 0.
\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = -\frac{1_{(\ell=i)}}{\hat{y}_\ell}
\]

We can now talk about the gradient w.r.t. the vector \(\hat{y}\)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\
\vdots
\end{bmatrix}
\]

\[
\cdot \cdot \cdot
\]

\[
\cdot \cdot \cdot
\]
\[ \frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = - \frac{1}{\hat{y}_\ell} \]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[ \nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k} \end{bmatrix} = - \begin{bmatrix} \hat{y}_\ell \\ \vdots \\ \hat{y}_\ell \end{bmatrix} e(\ell), \]

where \( e(\ell) \) is a \( k \)-dimensional vector whose \( \ell \)-th element is 1 and all other elements are 0.
\[ \frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = -\frac{1}{\hat{y}_\ell} \]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k} \end{bmatrix} = -\frac{1}{\hat{y}_\ell}
\]
\[
\frac{\partial}{\partial \hat{y}_i} \left( \mathcal{L}(\theta) \right) = -\frac{1_{(\ell=i)}}{\hat{y}_\ell}
\]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k}
\end{bmatrix} = -\frac{1}{\hat{y}_\ell}
\]

where \( e(\ell) \) is a \( k \)-dimensional vector whose \( \ell \)-th element is 1 and all other elements are 0.
\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = -\frac{1}{\hat{y}_\ell}
\]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k}
\end{bmatrix} = -\frac{1}{\hat{y}_\ell} \begin{bmatrix}
1_{\ell=1} \\
\vdots \\
1_{\ell=1}
\end{bmatrix}
\]
\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = -\frac{1_{(\ell=i)}}{\hat{y}_\ell}
\]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k}
\end{bmatrix} = -\frac{1}{\hat{y}_\ell} \begin{bmatrix}
1_{\ell=1} \\
1_{\ell=2} \\
\vdots \\
1_{\ell=k}
\end{bmatrix}
\]
\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = -\frac{1}{\hat{y}_\ell}
\]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k}
\end{bmatrix} = -\frac{1}{\hat{y}_\ell} \begin{bmatrix}
1_{\ell=1} \\
1_{\ell=2} \\
\vdots
\end{bmatrix}
\]

where \( e(\ell) \) is a \( k \)-dimensional vector whose \( \ell \)-th element is 1 and all other elements are 0.
\[
\frac{\partial}{\partial \hat{y}_i} \left( \mathcal{L}(\theta) \right) = -\frac{1_{(\ell=i)}}{\hat{y}_\ell}
\]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k}
\end{bmatrix} = -\frac{1}{\hat{y}_\ell} \begin{bmatrix}
1_{\ell=1} \\
\ddots \\
1_{\ell=k}
\end{bmatrix}
\]
\[
\frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = -\frac{1}_{(\ell=i)} \frac{\hat{y}_\ell}{\hat{y}_i}
\]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k}
\end{bmatrix} = -\frac{1}{\hat{y}_\ell} \begin{bmatrix}
1_{\ell=1} \\
\vdots \\
1_{\ell=k}
\end{bmatrix} = -\frac{1}{\hat{y}_\ell} e_\ell
\]
\[ \frac{\partial}{\partial \hat{y}_i} (\mathcal{L}(\theta)) = -\frac{1}{\hat{y}_\ell} \]

We can now talk about the gradient w.r.t. the vector \( \hat{y} \)

\[
\nabla_{\hat{y}} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_1} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}_k} \end{bmatrix} = -\frac{1}{\hat{y}_\ell} \begin{bmatrix} 1_{\ell=1} \\ 1_{\ell=2} \\ \vdots \\ 1_{\ell=k} \end{bmatrix} = -\frac{1}{\hat{y}_\ell} e_\ell
\]

where \( e(\ell) \) is a \( k \)-dimensional vector whose \( \ell \)-th element is 1 and all other elements are 0.
What we are actually interested in is

\[
\frac{\partial L(\theta)}{\partial a_{Li}} = \frac{\partial (- \log \hat{y}_\ell)}{\partial a_{Li}}
\]

Does \( \hat{y}_\ell \) depend on \( a_{Li} \)?

Indeed, it does.

\[
\hat{y}_\ell = \exp(\sum_i a_{Li}) \sum_i \exp(a_{Li})
\]

Having established this, we will now derive the full expression on the next slide.
What we are actually interested in is

\[
\frac{\partial \mathcal{L}(\theta)}{\partial a_{Li}} = \frac{\partial (- \log \hat{y}_\ell)}{\partial a_{Li}} = \frac{\partial (- \log \hat{y}_\ell)}{\partial \hat{y}_\ell} \frac{\partial \hat{y}_\ell}{\partial a_{Li}}
\]

Does \( \hat{y}_\ell \) depend on \( a_{Li} \)?

Indeed, it does.

\[
\hat{y}_\ell = \exp(\alpha L_\ell) \sum_i \exp(\alpha L_i)
\]

Having established this, we will now derive the full expression on the next slide.
What we are actually interested in is

\[
\frac{\partial \mathcal{L}(\theta)}{\partial a_{Li}} = \frac{\partial (- \log \hat{y}_\ell)}{\partial a_{Li}} = \frac{\partial (- \log \hat{y}_\ell)}{\partial \hat{y}_\ell} \frac{\partial \hat{y}_\ell}{\partial a_{Li}}
\]

Does \( \hat{y}_\ell \) depend on \( a_{Li} \)?
What we are actually interested in is

\[
\frac{\partial L(\theta)}{\partial a_{Li}} = \frac{\partial (-\log \hat{y}_\ell)}{\partial a_{Li}} = \frac{\partial (-\log \hat{y}_\ell)}{\partial \hat{y}_\ell} \frac{\partial \hat{y}_\ell}{\partial a_{Li}}
\]

Does \( \hat{y}_\ell \) depend on \( a_{Li} \)? Indeed, it does.
What we are actually interested in is

$$\frac{\partial \mathcal{L}(\theta)}{\partial a_{Li}} = \frac{\partial(- \log \hat{y}_\ell)}{\partial a_{Li}}$$

$$= \frac{\partial(- \log \hat{y}_\ell)}{\partial \hat{y}_\ell} \frac{\partial \hat{y}_\ell}{\partial a_{Li}}$$

Does $\hat{y}_\ell$ depend on $a_{Li}$? Indeed, it does.

$$\hat{y}_\ell = \frac{\exp(a_{L\ell})}{\sum_i \exp(a_{Li})}$$
What we are actually interested in is
\[
\frac{\partial L(\theta)}{\partial a_{Li}} = \frac{\partial (- \log \hat{y}_\ell)}{\partial a_{Li}} = \frac{\partial (- \log \hat{y}_\ell)}{\partial \hat{y}_\ell} \frac{\partial \hat{y}_\ell}{\partial a_{Li}}
\]

Does \( \hat{y}_\ell \) depend on \( a_{Li} \)? Indeed, it does.

\[
\hat{y}_\ell = \frac{\exp(a_{L\ell})}{\sum_i \exp(a_{Li})}
\]

Having established this, we will now derive the full expression on the next slide
\[
\frac{\partial}{\partial a_L} - \log \hat{y}_\ell =
\]
\[ \frac{\partial}{\partial a_{Li}} \log \hat{y}_\ell = - \frac{1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \hat{y}_\ell \]
\[
\frac{\partial}{\partial a_{Li}} \log \hat{y}_\ell = \frac{-1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \hat{y}_\ell = \frac{-1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \text{softmax}(a_L)_\ell
\]
\[
\frac{\partial}{\partial a_{LI}} \log \hat{y}_l = -\frac{1}{\hat{y}_l} \frac{\partial}{\partial a_{LI}} \hat{y}_l \\
= -\frac{1}{\hat{y}_l} \frac{\partial}{\partial a_{LI}} \text{softmax}(a_L)_l \\
= -\frac{1}{\hat{y}_l} \frac{\partial}{\partial a_{LI}} \frac{\exp(a_L)_l}{\sum_{i'} \exp(a_L)_l} 
\]
\[
\frac{\partial}{\partial a_{Li}} - \log \hat{y}_l = \frac{-1}{\hat{y}_l} \frac{\partial}{\partial a_{Li}} \hat{y}_l
\]

\[
= \frac{-1}{\hat{y}_l} \frac{\partial}{\partial a_{Li}} \text{softmax}(a_L)_l
\]

\[
= \frac{-1}{\hat{y}_l} \frac{\partial}{\partial a_{Li}} \exp(a_L)_l \sum_{i'} \exp(a_L)_l
\]

\[
\frac{\partial g(x)}{h(x)} = \frac{\partial g(x)}{\partial x} \frac{1}{h(x)} - \frac{g(x)}{h(x)^2} \frac{\partial h(x)}{\partial x}
\]
\[
\frac{\partial}{\partial a_{Li}} - \log \hat{y}_\ell = -\frac{1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \hat{y}_\ell \\
= -\frac{1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \text{softmax}(a_L)_\ell \\
= -\frac{1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \frac{\exp(a_L)_\ell}{\sum_{i'} \exp(a_L)_{i'}} \\
= -\frac{1}{\hat{y}_\ell} \left( \frac{\partial}{\partial a_{Li}} \exp(a_L)_{\ell} - \frac{\exp(a_L)_\ell \left( \frac{\partial}{\partial a_{Li}} \sum_{i'} \exp(a_L)_{i'} \right)}{\left( \sum_{i'} \exp(a_L)_{i'} \right)^2} \right)
\]

\[
\frac{\partial g(x)}{\partial x} = \frac{\partial g(x)}{\partial x} \frac{1}{h(x)} - \frac{g(x)}{h(x)^2} \frac{\partial h(x)}{\partial x}
\]
\[
\frac{\partial}{\partial a_L} \log \hat{y}_l = -1 \frac{\partial}{\partial a_L} \hat{y}_l
\]

\[
= -1 \frac{\partial}{\partial a_L} \text{softmax}(a_L)_l
\]

\[
= -1 \frac{\partial}{\partial a_L} \frac{\exp(a_L)_l}{\sum_{i'} \exp(a_L)_l}
\]

\[
= -1 \left( \frac{\partial}{\partial a_L} \exp(a_L)_l - \exp(a_L)_l \left( \frac{\partial}{\partial a_L} \sum_{i'} \exp(a_L)_{i'} \right) \right)
\]

\[
= -1 \left( \frac{1}_{(l=i)} \exp(a_L)_l - \exp(a_L)_l \left( \frac{\exp(a_L)_l}{\sum_{i'} \exp(a_L)_{i'}} \right) \right)
\]
\[
\frac{\partial}{\partial a_{Li}} \log \hat{y}_l = \frac{-1}{\hat{y}_l} \frac{\partial}{\partial a_{Li}} \hat{y}_l
\]
\[
= \frac{-1}{\hat{y}_l} \frac{\partial}{\partial a_{Li}} \text{softmax}(a_L)_l
\]
\[
= \frac{-1}{\hat{y}_l} \frac{\partial}{\partial a_{Li}} \frac{\exp(a_L)_l}{\sum_{i'} \exp(a_L)_{i'}}
\]
\[
= \frac{-1}{\hat{y}_l} \left( \frac{\partial}{\partial a_{Li}} \frac{\exp(a_L)_l}{\sum_{i'} \exp(a_L)_{i'}} - \frac{\exp(a_L)_l}{\sum_{i'} \exp(a_L)_{i'}} \right)
\]
\[
= \frac{-1}{\hat{y}_l} \left( \frac{1_{(l=i)}}{\sum_{i'} \exp(a_L)_{i'}} - \frac{\exp(a_L)_l}{\sum_{i'} \exp(a_L)_{i'}} \frac{\exp(a_L)_i}{\sum_{i'} \exp(a_L)_{i'}} \right)
\]
\[
= \frac{-1}{\hat{y}_l} \left( 1_{(l=i)} \text{softmax}(a_L)_l - \text{softmax}(a_L)_l \text{softmax}(a_L)_i \right)
\]
\[
\frac{\partial g(x)}{h(x)} = \frac{\partial g(x)}{\partial x} \frac{1}{h(x)} - g(x) \frac{\partial h(x)}{\partial x}
\]
\[
\frac{\partial}{\partial a_{Li}} - \log \hat{y}_\ell = \frac{-1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \hat{y}_\ell = \frac{-1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \text{softmax}(a_L)_\ell = \frac{-1}{\hat{y}_\ell} \frac{\partial}{\partial a_{Li}} \frac{\exp(a_L)_\ell}{\sum_{i'} \exp(a_L)_{i'}}
\]
\[
= \frac{-1}{\hat{y}_\ell} \left( \frac{\partial}{\partial a_{Li}} \exp(a_L)_\ell - \frac{\exp(a_L)_\ell \left( \frac{\partial}{\partial a_{Li}} \sum_{i'} \exp(a_L)_{i'} \right)}{\left( \sum_{i'} \left( \exp(a_L)_{i'} \right)^2 \right)} \right)
\]
\[
= \frac{-1}{\hat{y}_\ell} \left( \frac{1}{\ell = i} \frac{\exp(a_L)_\ell}{\sum_{i'} \exp(a_L)_{i'}} - \frac{\exp(a_L)_\ell \exp(a_L)_i}{\sum_{i'} \exp(a_L)_{i'} \sum_{i'} \exp(a_L)_{i'}} \right)
\]
\[
= \frac{-1}{\hat{y}_\ell} \left( \frac{1}{\ell = i} \text{softmax}(a_L)_\ell - \text{softmax}(a_L)_\ell \text{softmax}(a_L)_i \right)
\]
\[
= \frac{-1}{\hat{y}_\ell} \left( \frac{1}{\ell = i} \hat{y}_\ell - \hat{y}_\ell \hat{y}_i \right)
\]
\[
\frac{\partial}{\partial a_{L \ell}} \log \hat{y}_\ell = -\frac{1}{\hat{y}_\ell} \frac{\partial}{\partial a_{L \ell}} \hat{y}_\ell \\
= -\frac{1}{\hat{y}_\ell} \frac{\partial}{\partial a_{L \ell}} \text{softmax}(a_L)_\ell \\
= -\frac{1}{\hat{y}_\ell} \frac{\partial}{\partial a_{L \ell}} \frac{\exp(a_L)_\ell}{\sum_{i'} \exp(a_L)_{i'}} \\
= -\frac{1}{\hat{y}_\ell} \left( \frac{\partial}{\partial a_{L \ell}} \exp(a_L)_\ell - \frac{\exp(a_L)_\ell \left( \frac{\partial}{\partial a_{L \ell}} \sum_{i'} \exp(a_L)_{i'} \right)}{\left( \sum_{i'} \exp(a_L)_{i'} \right)^2} \right) \\
= -\frac{1}{\hat{y}_\ell} \left( \mathbb{1}_{(\ell = i)} \exp(a_L)_\ell - \frac{\exp(a_L)_\ell}{\sum_{i'} \exp(a_L)_{i'}} \frac{\exp(a_L)_i}{\sum_{i'} \exp(a_L)_{i'}} \right) \\
= -\frac{1}{\hat{y}_\ell} \left( \mathbb{1}_{(\ell = i)} \text{softmax}(a_L)_\ell - \text{softmax}(a_L)_\ell \text{softmax}(a_L)_i \right) \\
= -\frac{1}{\hat{y}_\ell} \left( \mathbb{1}_{(\ell = i)} \hat{y}_\ell - \hat{y}_\ell \hat{y}_i \right) \\
= -\left( \mathbb{1}_{(\ell = i)} - \hat{y}_i \right)
\]
So far we have derived the partial derivative w.r.t. the $i$-th element of $a_L$

$$\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,i}} = -(1_{\ell=i} - \hat{y}_i)$$

We can now write the gradient w.r.t. the vector $a_L$
So far we have derived the partial derivative w.r.t. the $i$-th element of $a_L$

$$\frac{\partial L(\theta)}{\partial a_{L,i}} = -(1_{\ell=i} - \hat{y}_i)$$

We can now write the gradient w.r.t. the vector $a_L$

$$\nabla_{a_L} L(\theta)$$
So far we have derived the partial derivative w.r.t. the \( i \)-th element of \( \mathbf{a}_L \)

\[
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,i}} = -(\mathbb{1}_{\ell=i} - \hat{y}_i)
\]

We can now write the gradient w.r.t. the vector \( \mathbf{a}_L \)

\[
\nabla_{\mathbf{a}_L} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,k}}
\end{bmatrix}
\]

\[-\log \hat{y}_\ell\]
So far we have derived the partial derivative w.r.t. the $i$-th element of $\mathbf{a}_L$

$$\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,i}} = -(1_{\ell=i} - \hat{y}_i)$$

We can now write the gradient w.r.t. the vector $\mathbf{a}_L$

$$\nabla_{\mathbf{a}_L} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{L1}} \\ \vdots \end{bmatrix}$$
So far we have derived the partial derivative w.r.t. the $i$-th element of $\mathbf{a}_L$

$$\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,i}} = -(1_{\ell=i} - \hat{y}_i)$$

We can now write the gradient w.r.t. the vector $\mathbf{a}_L$

$$\nabla_{\mathbf{a}_L} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{Lk}}
\end{bmatrix} = -\mathbf{e}(\ell) + \mathbf{\hat{y}}$$
So far we have derived the partial derivative w.r.t. the \( i \)-th element of \( a_L \)

\[
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,i}} = -(1_{\ell=i} - \hat{y}_i)
\]

We can now write the gradient w.r.t. the vector \( a_L \)

\[
\nabla_{a_L} \mathcal{L}(\theta) = \left[ \begin{array}{c}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{Lk}}
\end{array} \right] = \left[ \begin{array}{c}
-(1_{\ell=1} - \hat{y}_1) \\
\vdots \\
-(1_{\ell=k} - \hat{y}_k)
\end{array} \right] = -\left[ \begin{array}{c}
e^{\ell} - \hat{y}_\ell
\end{array} \right]
\]

\[x_1\quad x_2\quad x_n\]

\[W_1\quad h_1\quad a_1\]

\[W_2\quad h_2\quad a_2\]

\[W_3\quad h_3\quad a_3\]

\[-\log \hat{y}_\ell\]
So far we have derived the partial derivative w.r.t. the $i$-th element of $a_L$

$$\frac{\partial L(\theta)}{\partial a_{L,i}} = -(1_{\ell=i} - \hat{y}_i)$$

We can now write the gradient w.r.t. the vector $a_L$

$$\nabla_{a_L} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial a_{L1}} \\ \vdots \\ \frac{\partial L(\theta)}{\partial a_{Lk}} \end{bmatrix} = \begin{bmatrix} -(1_{\ell=1} - \hat{y}_1) \\ \vdots \\ -(1_{\ell=k} - \hat{y}_k) \end{bmatrix}$$
So far we have derived the partial derivative w.r.t. the $i$-th element of $a_L$

$$\frac{\partial L(\theta)}{\partial a_{L,i}} = -(\mathbb{1}_{\ell=i} - \hat{y}_i)$$

We can now write the gradient w.r.t. the vector $a_L$

$$\nabla_{a_L} L(\theta) = \left[ \begin{array}{c} \frac{\partial L(\theta)}{\partial a_{L1}} \\ \vdots \\ \frac{\partial L(\theta)}{\partial a_{Lk}} \end{array} \right] = \left[ \begin{array}{c} -(\mathbb{1}_{\ell=1} - \hat{y}_1) \\ -(\mathbb{1}_{\ell=2} - \hat{y}_2) \end{array} \right]$$
So far we have derived the partial derivative w.r.t. the $i$-th element of $a_L$

\[
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,i}} = -(\mathbb{1}_{\ell=i} - \hat{y}_i)
\]

We can now write the gradient w.r.t. the vector $a_L$

\[
\nabla_{a_L} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{Lk}}
\end{bmatrix} = \begin{bmatrix}
-(\mathbb{1}_{\ell=1} - \hat{y}_1) \\
-(\mathbb{1}_{\ell=2} - \hat{y}_2) \\
\vdots
\end{bmatrix}
\]
So far we have derived the partial derivative w.r.t. the $i$-th element of $a_L$

$$\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,i}} = -(\mathbb{1}_{\ell=i} - \hat{y}_i)$$

We can now write the gradient w.r.t. the vector $a_L$

$$\nabla_{a_L} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{L1}} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_{Lk}} \end{bmatrix} = \begin{bmatrix} -(\mathbb{1}_{\ell=1} - \hat{y}_1) \\ -(\mathbb{1}_{\ell=2} - \hat{y}_2) \\ \vdots \\ -(\mathbb{1}_{\ell=k} - \hat{y}_k) \end{bmatrix}$$

$$-\log \hat{y}_\ell$$
So far we have derived the partial derivative w.r.t. the $i$-th element of $\mathbf{a}_L$

$$\frac{\partial \mathcal{L}(\theta)}{\partial a_{L,i}} = -(1_{\ell=i} - \hat{y}_i)$$

We can now write the gradient w.r.t. the vector $\mathbf{a}_L$

$$\nabla_{\mathbf{a}_L} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L_1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{L_k}}
\end{bmatrix} = \begin{bmatrix}
-(1_{\ell=1} - \hat{y}_1) \\
-(1_{\ell=2} - \hat{y}_2) \\
\vdots \\
-(1_{\ell=k} - \hat{y}_k)
\end{bmatrix} = -(\mathbf{e}(\ell) - \hat{\mathbf{y}})$$
Module 4.6: Backpropagation: Computing Gradients w.r.t. Hidden Units
Quantities of interest (roadmap for the remaining part):

- Gradient w.r.t. output units
- Gradient w.r.t. hidden units
- Gradient w.r.t. weights and biases

\[ \frac{\partial \mathcal{L}(\theta)}{\partial W_{111}} = \frac{\partial \mathcal{L}(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_{111}} \]

- Talk to the weight directly
- Talk to the output layer
- Talk to the previous hidden layer
- Talk to the previous hidden layer
- and now talk to the weights

Our focus is on *Cross entropy loss* and *Softmax* output.
Chain rule along multiple paths: If a function $p(z)$ can be written as a function of intermediate results $q_i(z)$ then we have:

$$\frac{\partial p(z)}{\partial z} = \sum_m \frac{\partial p(z)}{\partial q_m(z)} \frac{\partial q_m(z)}{\partial z}$$

In our case:

$p(z)$ is the loss function $L(\theta)$

$z = h_{ij}$

$q_m(z) = a_{Lm}$
Chain rule along multiple paths: If a function $p(z)$ can be written as a function of intermediate results $q_i(z)$ then we have:

$$\frac{\partial p(z)}{\partial z} = \sum_m \frac{\partial p(z)}{\partial q_m(z)} \frac{\partial q_m(z)}{\partial z}$$
Chain rule along multiple paths: If a function $p(z)$ can be written as a function of intermediate results $q_i(z)$ then we have:

$$\frac{\partial p(z)}{\partial z} = \sum_m \frac{\partial p(z)}{\partial q_m(z)} \frac{\partial q_m(z)}{\partial z}$$

In our case:

- $p(z)$ is the loss function $\mathcal{L}(\theta)$
Chain rule along multiple paths: If a function $p(z)$ can be written as a function of intermediate results $q_i(z)$ then we have:

$$\frac{\partial p(z)}{\partial z} = \sum_m \frac{\partial p(z)}{\partial q_m(z)} \frac{\partial q_m(z)}{\partial z}$$

In our case:
- $p(z)$ is the loss function $\mathcal{L}(\theta)$
- $z = h_{ij}$
Chain rule along multiple paths: If a function $p(z)$ can be written as a function of intermediate results $q_i(z)$ then we have:

$$\frac{\partial p(z)}{\partial z} = \sum_m \frac{\partial p(z)}{\partial q_m(z)} \frac{\partial q_m(z)}{\partial z}$$

In our case:

- $p(z)$ is the loss function $\mathcal{L}(\theta)$
- $z = h_{ij}$
- $q_m(z) = a_{Lm}$
\[ \frac{\partial L(\theta)}{\partial h_{ij}} = k \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} \]

Now consider these two vectors, \( \nabla a_{i+1} L(\theta) \) and \( W_{i+1} \cdot j \), where \( W_{i+1} \cdot j \) is the \( j \)-th column of \( W_{i+1} \); see that,

\[ (W_{i+1} \cdot j)^T \nabla a_{i+1} L(\theta) = k \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j} \]
\[ \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} \]

Now consider these two vectors,
\[ \nabla a_{i+1} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,1}} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,k}} \end{bmatrix}; \]
\[ W_{i+1} \cdot j \] is the \( j \)-th column of \( W_{i+1} \); see that,
\[ (W_{i+1} \cdot j)^T \nabla a_{i+1} \mathcal{L}(\theta) = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j} \]
\[
\frac{\partial L(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} \\
= \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]
\[ \frac{\partial L(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} \]

\[ = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j} \]

Now consider these two vectors,

\[ \nabla a_{i+1} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial a_{i+1,1}} \\ \vdots \\ \frac{\partial L(\theta)}{\partial a_{i+1,k}} \end{bmatrix} \]

\[ W_{i+1} \cdot j = \begin{bmatrix} W_{i+1,1,j} \\ \vdots \\ W_{i+1,k,j} \end{bmatrix} \]

is the \( j \)-th column of \( W_{i+1} \); see that,

\[ (W_{i+1} \cdot j)^{T} \nabla a_{i+1} L(\theta) = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j} \]
\[
\frac{\partial L(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} L(\theta) = \begin{bmatrix}
\vdots \\
\end{bmatrix}; W_{i+1, \cdot, j} = \begin{bmatrix}
\vdots \\
\end{bmatrix}
\]

\[
a_{i+1} = W_{i+1} h_{ij} + b_{i+1}
\]
\[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,1}} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,k}} \end{bmatrix}; W_{i+1,\cdot,j} = \begin{bmatrix} W_{i+1,1,j} \\ \vdots \\ W_{i+1,k,j} \end{bmatrix}
\]
\[
\frac{\partial L(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} L(\theta) = \begin{bmatrix}
\frac{\partial L(\theta)}{\partial a_{i+1,1}} \\
\end{bmatrix} ; W_{i+1, \cdot, j} = \begin{bmatrix}
W_{i+1,1,j} \\
\end{bmatrix}
\]
\[ \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} \]

\[ = \sum_{m=1}^{k} \partial \mathcal{L}(\theta)_{\partial a_{i+1,m}} W_{i+1,m,j} \]

Now consider these two vectors,

\[ \nabla_{a_{i+1}} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,1}} \\ \vdots \end{bmatrix} ; W_{i+1,\cdot,j} = \begin{bmatrix} W_{i+1,1,j} \\ \vdots \end{bmatrix} \]

\[ a_{i+1} = W_{i+1} h_{ij} + b_{i+1} \]
\[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}}
\]

\[
= \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,k}}
\end{bmatrix}; W_{i+1, \cdot,j} = \begin{bmatrix}
W_{i+1,1,j} \\
\vdots
\end{bmatrix}
\]
\[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,k}}
\end{bmatrix}; \quad W_{i+1,\cdot,j} = \begin{bmatrix}
W_{i+1,1,j} \\
\vdots \\
W_{i+1,k,j}
\end{bmatrix}
\]
\[
\frac{\partial L(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}}
\]
\[
= \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial a_{i+1,1}} \\ \vdots \\ \frac{\partial L(\theta)}{\partial a_{i+1,k}} \end{bmatrix}; W_{i+1, \cdot,j} = \begin{bmatrix} W_{i+1,1,j} \\ \vdots \\ W_{i+1,k,j} \end{bmatrix}
\]

\[
a_{i+1} = W_{i+1} h_{ij} + b_{i+1}
\]
\[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} \\
= \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,k}}
\end{bmatrix} ; \quad W_{i+1,\cdot,j} = \begin{bmatrix}
W_{i+1,1,j} \\
\vdots \\
W_{i+1,k,j}
\end{bmatrix}
\]

\(W_{i+1,\cdot,j}\) is the \(j\)-th column of \(W_{i+1}\);

\[
\begin{align*}
a_{i+1} &= W_{i+1} h_{ij} + b_{i+1}
\end{align*}
\]
\[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}}
\]

\[
= \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,k}}
\end{bmatrix};
W_{i+1,j} = \begin{bmatrix}
W_{i+1,1,j} \\
\vdots \\
W_{i+1,k,j}
\end{bmatrix}
\]

\(W_{i+1,j}\) is the \(j\)-th column of \(W_{i+1}\); see that,

\[
a_{i+1} = W_{i+1} h_{ij} + b_{i+1}
\]
\[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}
\]

Now consider these two vectors,

\[
\nabla_{a_{i+1}} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i+1,k}}
\end{bmatrix}; \quad W_{i+1,\cdot,j} = \begin{bmatrix}
W_{i+1,1,j} \\
\vdots \\
W_{i+1,k,j}
\end{bmatrix}
\]

\(W_{i+1,\cdot,j}\) is the \(j\)-th column of \(W_{i+1}\); see that,

\[
(W_{i+1,\cdot,j})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) = a_{i+1} = W_{i+1} h_{ij} + b_{i+1}
\]
$$\frac{\partial L(\theta)}{\partial h_{ij}} = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} \frac{\partial a_{i+1,m}}{\partial h_{ij}}$$

$$= \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}$$

Now consider these two vectors,

$$\nabla_{a_{i+1}} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial a_{i+1,1}} \\ \vdots \\ \frac{\partial L(\theta)}{\partial a_{i+1,k}} \end{bmatrix}; W_{i+1,\cdot,j} = \begin{bmatrix} W_{i+1,1,j} \\ \vdots \\ W_{i+1,k,j} \end{bmatrix}$$

$W_{i+1,\cdot,j}$ is the $j$-th column of $W_{i+1}$; see that,

$$(W_{i+1,\cdot,j})^T \nabla_{a_{i+1}} L(\theta) = \sum_{m=1}^{k} \frac{\partial L(\theta)}{\partial a_{i+1,m}} W_{i+1,m,j}$$

$$a_{i+1} = W_{i+1} h_{ij} + b_{i+1}$$
We have, 
\[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,..j})^T \nabla a_{i+1} \mathcal{L}(\theta)
\]
We have, \[ \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,j})^T \nabla a_{i+1} \mathcal{L}(\theta) \]

We can now write the gradient w.r.t. \( h_i \)

\[ \nabla_{h_i} \mathcal{L}(\theta) \]
We have,
\[ \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,j})^T \nabla a_{i+1} \mathcal{L}(\theta) \]

We can now write the gradient w.r.t. \( h_i \)

\[ \nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix} \partial \mathcal{L}(\theta) / \partial h_{11} \\ \partial \mathcal{L}(\theta) / \partial h_{12} \\ \vdots \\ \partial \mathcal{L}(\theta) / \partial h_{in} \end{bmatrix} = \begin{bmatrix} (W_{i+1,1})^T \nabla a_{i+1} \mathcal{L}(\theta) \\ (W_{i+1,2})^T \nabla a_{i+1} \mathcal{L}(\theta) \\ \vdots \\ (W_{i+1,n})^T \nabla a_{i+1} \mathcal{L}(\theta) \end{bmatrix} = (W_{i+1})^T \nabla a_{i+1} \mathcal{L}(\theta) \]
We have, \[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,..,j})^T \nabla a_{i+1} \mathcal{L}(\theta)
\]

We can now write the gradient w.r.t. \( h_i \)

\[
\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i2}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{in}}
\end{bmatrix} = \begin{bmatrix}
(W_{i+1,1})^T \nabla a_{i+1} \mathcal{L}(\theta) \\
(W_{i+1,2})^T \nabla a_{i+1} \mathcal{L}(\theta) \\
\vdots \\
(W_{i+1,n})^T \nabla a_{i+1} \mathcal{L}(\theta)
\end{bmatrix}
\]
We have, \( \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,j})^T \nabla a_{i+1} \mathcal{L}(\theta) \)

We can now write the gradient w.r.t. \( h_i \)

\[
\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \end{bmatrix} = \begin{bmatrix} (W_{i+1,\cdot,1})^T \nabla a_{i+1} \mathcal{L}(\theta) \end{bmatrix}
\]
We have, \[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,..,j})^T \nabla_{a_{i+1}} \mathcal{L}(\theta)
\]

We can now write the gradient w.r.t. \( h_i \)

\[
\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i2}} \\
\end{bmatrix} = \begin{bmatrix}
(W_{i+1,..,1})^T \nabla_{a_{i+1}} \mathcal{L}(\theta)
\end{bmatrix}
\]
We have, \( \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,..,j})^T \nabla a_{i+1} \mathcal{L}(\theta) \)

We can now write the gradient w.r.t. \( h_i \)

\[
\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\ \frac{\partial \mathcal{L}(\theta)}{\partial h_{i2}} \end{bmatrix} = \begin{bmatrix} (W_{i+1,\cdot,1})^T \nabla a_{i+1} \mathcal{L}(\theta) \\ (W_{i+1,\cdot,2})^T \nabla a_{i+1} \mathcal{L}(\theta) \end{bmatrix}
\]
We have, \[ \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,j})^T \nabla a_{i+1} \mathcal{L}(\theta) \]

We can now write the gradient w.r.t. \( h_i \)

\[ \nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\ \frac{\partial \mathcal{L}(\theta)}{\partial h_{i2}} \\ \vdots \end{bmatrix} = \begin{bmatrix} (W_{i+1,1})^T \nabla a_{i+1} \mathcal{L}(\theta) \\ (W_{i+1,2})^T \nabla a_{i+1} \mathcal{L}(\theta) \\ \vdots \end{bmatrix} \]
We have, \( \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,..,j})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) \)

We can now write the gradient w.r.t. \( h_i \)

\[
\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i2}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{in}}
\end{bmatrix} = \begin{bmatrix}
(W_{i+1,..,1})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) \\
(W_{i+1,..,2})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) \\
\vdots
\end{bmatrix}
\]

We are almost done except that we do not know how to calculate \( \nabla_{a_{i+1}} \mathcal{L}(\theta) \) for \( i<L-1 \)
We have,\[
\frac{\partial L(\theta)}{\partial h_{ij}} = (W_{i+1,j})^T \nabla_{a_{i+1}} L(\theta)
\]

We can now write the gradient w.r.t. \( h_i \)

\[
\nabla_{h_i} L(\theta) = \begin{bmatrix}
\frac{\partial L(\theta)}{\partial h_{i1}} \\
\frac{\partial L(\theta)}{\partial h_{i2}} \\
\vdots \\
\frac{\partial L(\theta)}{\partial h_{in}}
\end{bmatrix} = \begin{bmatrix}
(W_{i+1,1})^T \nabla_{a_{i+1}} L(\theta) \\
(W_{i+1,2})^T \nabla_{a_{i+1}} L(\theta) \\
\vdots \\
(W_{i+1,n})^T \nabla_{a_{i+1}} L(\theta)
\end{bmatrix}
\]
We have, \[
\frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,j})^T \nabla_{a_{i+1}} \mathcal{L}(\theta)
\]

We can now write the gradient w.r.t. \( h_i \)

\[
\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i2}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{in}}
\end{bmatrix} = \begin{bmatrix}
(W_{i+1,1})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) \\
(W_{i+1,2})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) \\
\vdots \\
(W_{i+1,n})^T \nabla_{a_{i+1}} \mathcal{L}(\theta)
\end{bmatrix}
\]

\[
= (W_{i+1})^T (\nabla_{a_{i+1}} \mathcal{L}(\theta))
\]
We have, \( \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,:j})^T \nabla a_{i+1} \mathcal{L}(\theta) \)

We can now write the gradient w.r.t. \( h_i \)

\[
\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i2}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{in}}
\end{bmatrix} = \begin{bmatrix}
(W_{i+1,:1})^T \nabla a_{i+1} \mathcal{L}(\theta) \\
(W_{i+1,:2})^T \nabla a_{i+1} \mathcal{L}(\theta) \\
\vdots \\
(W_{i+1,:n})^T \nabla a_{i+1} \mathcal{L}(\theta)
\end{bmatrix} = (W_{i+1})^T (\nabla a_{i+1} \mathcal{L}(\theta))
\]

We are almost done except that we do not know how to calculate \( \nabla a_{i+1} \mathcal{L}(\theta) \) for \( i < L - 1 \)
We have, $rac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} = (W_{i+1,j})^T \nabla_{a_{i+1}} \mathcal{L}(\theta)$

We can now write the gradient w.r.t. $h_i$

$$
\nabla_{h_i} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{i2}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial h_{in}}
\end{bmatrix} = \begin{bmatrix}
(W_{i+1,1})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) \\
(W_{i+1,2})^T \nabla_{a_{i+1}} \mathcal{L}(\theta) \\
\vdots \\
(W_{i+1,n})^T \nabla_{a_{i+1}} \mathcal{L}(\theta)
\end{bmatrix}
$$

$$
= (W_{i+1})^T (\nabla_{a_{i+1}} \mathcal{L}(\theta))
$$

- We are almost done except that we do not know how to calculate $\nabla_{a_{i+1}} \mathcal{L}(\theta)$ for $i < L - 1$
- We will see how to compute that
\[ \nabla_{a_i} \mathcal{L}(\theta) \]
\[ \nabla_{a_i} \mathcal{L}(\theta) = \begin{bmatrix} \partial \mathcal{L}(\theta) \partial a_{i1} \\ \vdots \\ \partial \mathcal{L}(\theta) \partial a_{in} \end{bmatrix} \]
\( \nabla_{a_i} \mathcal{L} (\theta) = \begin{bmatrix} \frac{\partial \mathcal{L} (\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial \mathcal{L} (\theta)}{\partial a_{in}} \end{bmatrix} \)
\[ \nabla_{a_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i1}} \\ \vdots \end{bmatrix} \]
\[ \nabla_{a_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_{in}} \end{bmatrix} \]
\[ \nabla_{a_i} \mathcal{L}(\theta) = \left[ \frac{\partial \mathcal{L}(\theta)}{\partial a_{i1}} \right. \]

\[ \vdots \]

\[ \left. \frac{\partial \mathcal{L}(\theta)}{\partial a_{in}} \right] \]
\( \nabla_{ai} L(\theta) = \begin{bmatrix}
\frac{\partial L(\theta)}{\partial a_{i1}} \\
\vdots \\
\frac{\partial L(\theta)}{\partial a_{in}}
\end{bmatrix} \)

\[
\frac{\partial L(\theta)}{\partial a_{ij}} = \frac{\partial L(\theta)}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial a_{ij}}
\]
\[
\n\nabla_{a_i} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{i1}} \\
\vdots \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{in}}
\end{bmatrix}
\]

\[
\frac{\partial \mathcal{L}(\theta)}{\partial a_{ij}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial a_{ij}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} g'(a_{ij}) \quad [\because h_{ij} = g(a_{ij})]
\]
\[ \nabla_{a_i} \mathcal{L}(\theta) = \left[ \begin{array} {c} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_{in}} \end{array} \right] \]

\[
\frac{\partial \mathcal{L}(\theta)}{\partial a_{ij}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial a_{ij}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} g'(a_{ij}) \quad [\because h_{ij} = g(a_{ij})] 
\]

\[ \nabla_{a_i} \mathcal{L}(\theta) \]
\[ \nabla_{a_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_{in}} \end{bmatrix} \]

\[ \frac{\partial \mathcal{L}(\theta)}{\partial a_{ij}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial a_{ij}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} g'(a_{ij}) \quad [\because h_{ij} = g(a_{ij})] \]

\[ \nabla_{a_i} \mathcal{L}(\theta) = \begin{bmatrix} \end{bmatrix} \]
\[ \nabla_{a_i} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial L(\theta)}{\partial a_{in}} \end{bmatrix} \]

\[ \frac{\partial L(\theta)}{\partial a_{ij}} = \frac{\partial L(\theta)}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial a_{ij}} = \frac{\partial L(\theta)}{\partial h_{ij}} g'(a_{ij}) \quad [\because h_{ij} = g(a_{ij})] \]

\[ \nabla_{a_i} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial h_{i1}} g'(a_{i1}) \\ \vdots \\ \frac{\partial L(\theta)}{\partial h_{in}} g'(a_{in}) \end{bmatrix} \]
\[ \nabla_{a_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_{in}} \end{bmatrix} \]

\[
\frac{\partial \mathcal{L}(\theta)}{\partial a_{ij}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial a_{ij}} = \frac{\partial \mathcal{L}(\theta)}{\partial h_{ij}} g'(a_{ij}) \quad [:: h_{ij} = g(a_{ij})] 
\]

\[
\nabla_{a_i} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial h_{i1}} g'(a_{i1}) \\ \vdots \\ \end{bmatrix}
\]
\[ \nabla_{a_{i}}L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial L(\theta)}{\partial a_{in}} \end{bmatrix} \]

\[ \frac{\partial L(\theta)}{\partial a_{ij}} = \frac{\partial L(\theta)}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial a_{ij}} \]

\[ = \frac{\partial L(\theta)}{\partial h_{ij}} g'(a_{ij}) \quad [\because h_{ij} = g(a_{ij})] \]

\[ \nabla_{a_{i}}L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial h_{i1}} g'(a_{i1}) \\ \vdots \\ \frac{\partial L(\theta)}{\partial h_{in}} g'(a_{in}) \end{bmatrix} \]
\[ \nabla_{a_i} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial a_{i1}} \\ \vdots \\ \frac{\partial L(\theta)}{\partial a_{in}} \end{bmatrix} \]

\[ \frac{\partial L(\theta)}{\partial a_{ij}} = \frac{\partial L(\theta)}{\partial h_{ij}} \frac{\partial h_{ij}}{\partial a_{ij}} = \frac{\partial L(\theta)}{\partial h_{ij}} g'(a_{ij}) \quad \text{[} \because h_{ij} = g(a_{ij}) \text{]} \]

\[ \nabla_{a_i} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial h_{i1}} g'(a_{i1}) \\ \vdots \\ \frac{\partial L(\theta)}{\partial h_{in}} g'(a_{in}) \end{bmatrix} \]

\[ = \nabla_{h_i} L(\theta) \odot [\ldots, g'(a_{ik}), \ldots] \]
Module 4.7: Backpropagation: Computing Gradients w.r.t. Parameters
Quantities of interest (roadmap for the remaining part):

- Gradient w.r.t. output units
- Gradient w.r.t. hidden units
- Gradient w.r.t. weights and biases

\[
\frac{\partial L(\theta)}{\partial W_{111}} = \frac{\partial L(\theta)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial h_2} \frac{\partial h_2}{\partial a_2} \frac{\partial a_2}{\partial h_1} \frac{\partial h_1}{\partial a_1} \frac{\partial a_1}{\partial W_{111}}
\]

Talk to the weight directly
Talk to the output layer
Talk to the previous hidden layer
Talk to the previous hidden layer
and now talk to the weights

- Our focus is on \textit{Cross entropy loss} and \textit{Softmax} output.
Recall that,

$$a_k = b_k + W_k h_{k-1}$$
Recall that,

\[ a_k = b_k + W_k h_{k-1} \]

\[ \frac{\partial a_{ki}}{\partial W_{kij}} = h_{k-1,j} \]
Recall that,

\[ a_k = b_k + W_k h_{k-1} \]

\[ \frac{\partial a_{ki}}{\partial W_{kij}} = h_{k-1,j} \]

\[ \frac{\partial L(\theta)}{\partial W_{kij}} \]
Recall that,

\[ a_k = b_k + W_k h_{k-1} \]

\[ \frac{\partial a_{ki}}{\partial W_{kij}} = h_{k-1,j} \]

\[ \frac{\partial L(\theta)}{\partial W_{kij}} = \frac{\partial L(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kij}} \]

\[ -\log \hat{y}_l \]
Recall that,

\[ a_k = b_k + W_k h_{k-1} \]

\[ \frac{\partial a_{ki}}{\partial W_{kij}} = h_{k-1,j} \]

\[ \frac{\partial L(\theta)}{\partial W_{kij}} = \frac{\partial L(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kij}} = \frac{\partial L(\theta)}{\partial a_{ki}} h_{k-1,j} \]
Recall that,

\[ a_k = b_k + W_k h_{k-1} \]

\[ \frac{\partial a_{ki}}{\partial W_{kj}} = h_{k-1,j} \]

\[ \frac{\partial \mathcal{L}(\theta)}{\partial W_{kj}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kj}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} h_{k-1,j} \]

\[ \nabla_{W_k} \mathcal{L}(\theta) = \]
Recall that,

\[ a_k = b_k + W_k h_{k-1} \]

\[ \frac{\partial a_{ki}}{\partial W_{kij}} = h_{k-1,j} \]

\[ \frac{\partial L(\theta)}{\partial W_{kij}} = \frac{\partial L(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kij}} = \frac{\partial L(\theta)}{\partial a_{ki}} h_{k-1,j} \]

\[ \nabla_{W_k} L(\theta) = \begin{bmatrix}
\frac{\partial L(\theta)}{\partial W_{k11}} & \frac{\partial L(\theta)}{\partial W_{k12}} & \cdots & \frac{\partial L(\theta)}{\partial W_{k1n}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial L(\theta)}{\partial W_{knn}}
\end{bmatrix} \]
Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like
Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like

$$
\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k11}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k12}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k13}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k21}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k22}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k23}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k31}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k32}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k33}}
\end{bmatrix} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{kj}}{\partial W_{kij}}
$$
Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like

$$\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k11}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k12}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k13}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k21}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k22}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k23}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k31}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k32}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k33}}
\end{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial W_{ki}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{ki}}$$

$$\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,3}
\end{bmatrix} = $$
Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like

\[
\nabla_{W_k} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k11}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k12}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k13}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k21}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k22}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k23}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k31}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k32}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k33}}
\end{bmatrix}
\]

\[
\nabla_{W_k} \mathcal{L}(\theta) = \nabla_{a_k} \mathcal{L}(\theta) \cdot (h_{k-1})^T
\]

\[
\nabla_{W_k} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_k} h_{k-1,3}
\end{bmatrix}
\]
Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like

$$
\nabla_{W_k} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k11}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k12}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k13}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k21}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k22}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k23}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k31}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k32}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k33}}
\end{bmatrix}
$$

$$
\frac{\partial \mathcal{L}(\theta)}{\partial W_{kij}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kij}}
$$

$$
\nabla_{W_k} \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,3}
\end{bmatrix}
$$
Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like

$$
\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k11}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k12}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k13}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k21}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k22}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k23}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k31}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k32}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k33}}
\end{bmatrix}
\quad \frac{\partial \mathcal{L}(\theta)}{\partial W_{kij}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kij}}
$$

$$
\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,3}
\end{bmatrix}
= \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,3}
\end{bmatrix}
$$
Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like

$$
\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k11}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k12}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k13}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k21}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k22}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k23}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k31}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k32}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k33}}
\end{bmatrix} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kij}}
$$

$$
\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,3}
\end{bmatrix} =
$$
Lets take a simple example of a $W_k \in \mathbb{R}^{3 \times 3}$ and see what each entry looks like

\[
\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k11}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k12}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k13}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k21}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k22}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k23}} \\
\frac{\partial \mathcal{L}(\theta)}{\partial W_{k31}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k32}} & \frac{\partial \mathcal{L}(\theta)}{\partial W_{k33}}
\end{bmatrix}
\]

\[
\nabla W_k \mathcal{L}(\theta) = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial W_{kij}}
\]

\[
\nabla W_k \mathcal{L}(\theta) = \begin{bmatrix}
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k1}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k2}} h_{k-1,3} \\
\frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,1} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,2} & \frac{\partial \mathcal{L}(\theta)}{\partial a_{k3}} h_{k-1,3}
\end{bmatrix} = \nabla a_k \mathcal{L}(\theta) \cdot h_{k-1}^T
\]
Finally, coming to the biases

\[ b_k = b_k + \sum_j W_{kj} h_{k-1,j}, \]

\[ \partial L(\theta) \partial b_k = \partial L(\theta) \partial a_k \partial a_k \partial b_k = \partial L(\theta) \partial a_k \]

We can now write the gradient w.r.t. the vector

\[ \nabla b_k L(\theta) = \begin{bmatrix} \partial L(\theta) a_1 \\ \partial L(\theta) a_2 \\ \vdots \\ \partial L(\theta) a_n \end{bmatrix} = \nabla a_k L(\theta) \]
Finally, coming to the biases

\[ a_{ki} = b_{ki} + \sum_j W_{kij} h_{k-1,j} \]
Finally, coming to the biases

\[ a_{ki} = b_{ki} + \sum_j W_{kij} h_{k-1,j} \]

\[ \frac{\partial \mathcal{L}(\theta)}{\partial b_{ki}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial b_{ki}} \]
Finally, coming to the biases

\[ a_{ki} = b_{ki} + \sum_j W_{ki j} h_{k-1,j} \]

\[
\frac{\partial L(\theta)}{\partial b_{ki}} = \frac{\partial L(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial b_{ki}} = \frac{\partial L(\theta)}{\partial a_{ki}}
\]
Finally, coming to the biases

\[ a_{ki} = b_{ki} + \sum_j W_{ki j} h_{k-1,j} \]

\[ \frac{\partial \mathcal{L}(\theta)}{\partial b_{ki}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial b_{ki}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \]

We can now write the gradient w.r.t. the vector \( b_k \)
Finally, coming to the biases

\[ a_{ki} = b_{ki} + \sum_j W_{kj} h_{k-1,j}, \]

\[ \frac{\partial \mathcal{L}(\theta)}{\partial b_{ki}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial b_{ki}} = \frac{\partial \mathcal{L}(\theta)}{\partial a_{ki}} \]

We can now write the gradient w.r.t. the vector \( b_k \)

\[ \nabla_{b_k} \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta)}{\partial a_1} \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_2} \\ \vdots \\ \frac{\partial \mathcal{L}(\theta)}{\partial a_n} \end{bmatrix} \]
Finally, coming to the biases

$$a_{ki} = b_{ki} + \sum_j W_{kij} h_{k-1,j}$$

$$\frac{\partial L(\theta)}{\partial b_{ki}} = \frac{\partial L(\theta)}{\partial a_{ki}} \frac{\partial a_{ki}}{\partial b_{ki}} = \frac{\partial L(\theta)}{\partial a_{ki}}$$

We can now write the gradient w.r.t. the vector $b_k$

$$\nabla_{b_k} L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{a_{k1}} \\ \frac{\partial L(\theta)}{a_{k2}} \\ \vdots \\ \frac{\partial L(\theta)}{a_{kn}} \end{bmatrix} = \nabla_{a_k} L(\theta)$$
Module 4.8: Backpropagation: Pseudo code
Finally, we have all the pieces of the puzzle
Finally, we have all the pieces of the puzzle

\[ \nabla_{a_L} L(\theta) \] (gradient w.r.t. output layer)

\[ \nabla_{a_k} L(\theta), \nabla_{a_k} L(\theta) \] (gradient w.r.t. hidden layers, \( 1 \leq k < L \))

\[ \nabla_{W_k} L(\theta), \nabla_{b_k} L(\theta) \] (gradient w.r.t. weights and biases, \( 1 \leq k \leq L \))
Finally, we have all the pieces of the puzzle

$$\nabla_{a_L} \mathcal{L}(\theta) \quad \text{(gradient w.r.t. output layer)}$$

$$\nabla_{h_k} \mathcal{L}(\theta), \nabla_{a_k} \mathcal{L}(\theta) \quad \text{(gradient w.r.t. hidden layers, } 1 \leq k < L)$$
Finally, we have all the pieces of the puzzle

$$\nabla_{a_L} \mathcal{L}(\theta) \quad \text{(gradient w.r.t. output layer)}$$

$$\nabla_{h_k} \mathcal{L}(\theta), \nabla_{a_k} \mathcal{L}(\theta) \quad \text{(gradient w.r.t. hidden layers, } 1 \leq k < L)$$

$$\nabla_{W_k} \mathcal{L}(\theta), \nabla_{b_k} \mathcal{L}(\theta) \quad \text{(gradient w.r.t. weights and biases, } 1 \leq k \leq L)$$
Finally, we have all the pieces of the puzzle

$$\nabla_{a_L} \mathcal{L}(\theta)$$ (gradient w.r.t. output layer)

$$\nabla_{h_k} \mathcal{L}(\theta), \nabla_{a_k} \mathcal{L}(\theta)$$ (gradient w.r.t. hidden layers, $1 \leq k < L$)

$$\nabla_{W_k} \mathcal{L}(\theta), \nabla_{b_k} \mathcal{L}(\theta)$$ (gradient w.r.t. weights and biases, $1 \leq k \leq L$)

We can now write the full learning algorithm
Algorithm: gradient_descent()

\[ t \leftarrow 0; \]
\[ \text{max\_iterations} \leftarrow 1000; \]
\[ \text{Initialize } \theta_0 = [W^0_1, \ldots, W^0_L, b^0_1, \ldots, b^0_L]; \]
**Algorithm:** gradient_descent()

$t \leftarrow 0;$

$\text{max}_{}\text{iterations} \leftarrow 1000;$

$\text{Initialize} \quad \theta_0 = [W_1^0, ..., W_L^0, b_1^0, ..., b_L^0];$

\[\text{while } t++ < \text{max}_{}\text{iterations } \text{do}\]

end
Algorithm: gradient_descent()

$t \leftarrow 0$;
$max\_iterations \leftarrow 1000$;
Initialize $\theta_0 = [W^0_1, ..., W^0_L, b^0_1, ..., b^0_L]$;

while $t++ < max\_iterations$ do
    $h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, \hat{y} = forward\_propagation(\theta_t)$;

end
Algorithm: gradient_descent()

\[ \text{Initialize } \theta_0 = [W_0^0, ..., W_L^0, b_1^0, ..., b_L^0]; \]

\[
\text{while } t++ < \text{max\_iterations} \text{ do}
\]

\[
\begin{align*}
&h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, \hat{y} = \text{forward\_propagation}(\theta_t); \\
&\nabla \theta_t = \text{backward\_propagation}(h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y}); \\
\end{align*}
\]

end
Algorithm: gradient_descent()

$t \leftarrow 0$;
$max\_iterations \leftarrow 1000$;

Initialize $\theta_0 = [W_1^0, ..., W_L^0, b_1^0, ..., b_L^0]$;

while $t++ < max\_iterations$ do

$h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, \hat{y} = forward\_propagation(\theta_t)$;
$\nabla \theta_t = backward\_propagation(h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y})$;
$\theta_{t+1} \leftarrow \theta_t - \eta \nabla \theta_t$;

end
Algorithm: forward_propagation(θ)

for \( k = 1 \) to \( L - 1 \) do

\[ a_k = b_k + W_k h_{k-1}; \]

\[ h_k = g(a_k); \]

end

\[ a_L = b_L + W_L h_{L-1}; \]

\[ \hat{y} = O(a_L); \]
Algorithm: forward_propagation(θ)

for $k = 1$ to $L - 1$ do

end
Algorithm: forward_propagation($\theta$)

for $k = 1$ to $L - 1$ do
    $a_k = b_k + W_k h_{k-1};$
end
Algorithm: forward_propagation(\(\theta\))

for \(k = 1 \text{ to } L - 1\) do
  \[a_k = b_k + W_k h_{k-1};\]
  \[h_k = g(a_k);\]
end
Algorithm: forward_propagation($\theta$)

\begin{align*}
\text{for } k = 1 \text{ to } L - 1 \text{ do} \\
\quad & a_k = b_k + W_k h_{k-1}; \\
\quad & h_k = g(a_k); \\
\text{end} \\
\text{end}
\end{align*}
**Algorithm:** forward_propagation($\theta$)

\[
\text{for } k = 1 \text{ to } L - 1 \text{ do} \\
\quad a_k = b_k + W_k h_{k-1}; \\
\quad h_k = g(a_k); \\
\text{end} \\
\quad a_L = b_L + W_L h_{L-1}; \\
\hat{y} = O(a_L);
\]
Just do a forward propagation and compute all $h_i$’s, $a_i$’s, and $\hat{y}$

**Algorithm:** back_propagation($h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y}$)

//Compute output gradient ;
Just do a forward propagation and compute all \(h_i\)'s, \(a_i\)'s, and \(\hat{y}\)

**Algorithm:** back_propagation\((h_1, h_2, \ldots, h_{L-1}, a_1, a_2, \ldots, a_L, y, \hat{y})\)

// Compute output gradient:
\[
\nabla_a L(\theta) = -(e(y) - \hat{y})
\]
Just do a forward propagation and compute all $h_i$'s, $a_i$'s, and $\hat{y}$

**Algorithm:** `back_propagation($h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y}$)`

//Compute output gradient ;
$\nabla a_L \mathcal{L}(\theta) = -(e(y) - \hat{y})$ ;

for $k = L$ to 1 do

\[
\nabla h_k^{L-1}(\theta) = W_k^T \nabla a_k^L(\theta)
\]

// Compute gradients w.r.t. layer below (pre-activation);
$\nabla a_k^{L-1}(\theta) = \nabla h_k^{L-1}(\theta) \odot [g'(a_k^{L-1}, j), ...]$ ;

end
Just do a forward propagation and compute all $h_i$’s, $a_i$’s, and $\hat{y}$

**Algorithm:** back_propagation($h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y}$)

// Compute output gradient;
$\nabla a_L L(\theta) = -(e(y) - \hat{y})$;

for $k = L$ to 1 do
  // Compute gradients w.r.t. parameters;

end
Just do a forward propagation and compute all $h_i$’s, $a_i$’s, and $\hat{y}$

**Algorithm:** back_propagation($h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y}$)

// Compute output gradient ;
$\nabla_{a_L} \mathcal{L}(\theta) = -(e(y) - \hat{y})$ ;

for $k = L$ to 1 do

    // Compute gradients w.r.t. parameters ;
    $\nabla_{W_k} \mathcal{L}(\theta) = \nabla_{a_k} \mathcal{L}(\theta) h_{k-1}^T$ ;

end
Just do a forward propagation and compute all $h_i$'s, $a_i$'s, and $\hat{y}$

**Algorithm:** $\text{back\_propagation}(h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y})$

// Compute output gradient;
$\nabla_{a_L} \mathcal{L}(\theta) = -(e(y) - \hat{y})$

for $k = L$ to 1 do

    // Compute gradients w.r.t. parameters;
    $\nabla_{W_k} \mathcal{L}(\theta) = \nabla_{a_k} \mathcal{L}(\theta) h_{k-1}^T$
    $\nabla_{b_k} \mathcal{L}(\theta) = \nabla_{a_k} \mathcal{L}(\theta)$

end
Just do a forward propagation and compute all $h_i$'s, $a_i$'s, and $\hat{y}$

**Algorithm:** backpropagation($h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y}$)

// Compute output gradient;
$\nabla_{a_L} \mathcal{L}(\theta) = -(e(y) - \hat{y})$;

for $k = L$ to 1 do

    // Compute gradients w.r.t. parameters;
    $\nabla_{W_k} \mathcal{L}(\theta) = \nabla_{a_k} \mathcal{L}(\theta) h_{k-1}^T$;
    $\nabla_{b_k} \mathcal{L}(\theta) = \nabla_{a_k} \mathcal{L}(\theta)$;

    // Compute gradients w.r.t. layer below;
    $\nabla_{h_k} \mathcal{L}(\theta) = W_k^T (\nabla_{a_k} \mathcal{L}(\theta))$;
    $\nabla_{a_{k-1}} \mathcal{L}(\theta) = \nabla_{h_k} \mathcal{L}(\theta) \odot [\ldots, g'(a_{k-1}, j), \ldots]$;

end
Just do a forward propagation and compute all $h_i$'s, $a_i$'s, and $\hat{y}$

**Algorithm:** back_propagation($h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y}$)

// Compute output gradient;
$\nabla_{a_L} \mathcal{L}(\theta) = -(e(y) - \hat{y})$;

for $k = L$ to 1 do

    // Compute gradients w.r.t. parameters;
    $\nabla_{W_k} \mathcal{L}(\theta) = \nabla_{a_k} \mathcal{L}(\theta) h_{k-1}^T$;
    $\nabla_{b_k} \mathcal{L}(\theta) = \nabla_{a_k} \mathcal{L}(\theta)$;

    // Compute gradients w.r.t. layer below;
    $\nabla_{h_{k-1}} \mathcal{L}(\theta) = W_k^T (\nabla_{a_k} \mathcal{L}(\theta))$;

end
Just do a forward propagation and compute all $h_i$’s, $a_i$’s, and $\hat{y}$

**Algorithm:** back_propagation($h_1, h_2, \ldots, h_{L-1}, a_1, a_2, \ldots, a_L, y, \hat{y}$)

//Compute output gradient ;
$\nabla_{a_L} \mathcal{L} (\theta) = -(e(y) - \hat{y})$ ;

for $k = L$ to 1 do

// Compute gradients w.r.t. parameters ;
$\nabla_{W_k} \mathcal{L} (\theta) = \nabla_{a_k} \mathcal{L} (\theta) h_{k-1}^T$ ;
$\nabla_{b_k} \mathcal{L} (\theta) = \nabla_{a_k} \mathcal{L} (\theta)$ ;

// Compute gradients w.r.t. layer below ;
$\nabla_{h_{k-1}} \mathcal{L} (\theta) = W_k^T (\nabla_{a_k} \mathcal{L} (\theta))$ ;

// Compute gradients w.r.t. layer below (pre-activation);
Just do a forward propagation and compute all $h_i$'s, $a_i$'s, and $\hat{y}$

**Algorithm:** backPropagation($h_1, h_2, ..., h_{L-1}, a_1, a_2, ..., a_L, y, \hat{y}$)

```plaintext
// Compute output gradient
\n\n\n∇_{a_L} \mathcal{L}(\theta) = -(e(y) - \hat{y}) ;

for $k = L$ to 1 do

    // Compute gradients w.r.t. parameters
    ∇_{W_k} \mathcal{L}(\theta) = ∇_{a_k} \mathcal{L}(\theta) h_{k-1}^T ;
    ∇_{b_k} \mathcal{L}(\theta) = ∇_{a_k} \mathcal{L}(\theta) ;

    // Compute gradients w.r.t. layer below
    ∇_{h_{k-1}} \mathcal{L}(\theta) = W_k^T (∇_{a_k} \mathcal{L}(\theta)) ;

    // Compute gradients w.r.t. layer below (pre-activation)
    ∇_{a_{k-1}} \mathcal{L}(\theta) = ∇_{h_{k-1}} \mathcal{L}(\theta) \odot [\ldots, g'(a_{k-1,j}), \ldots] ;

end
```
Module 4.9: Derivative of the activation function
Now, the only thing we need to figure out is how to compute $g'$
Now, the only thing we need to figure out is how to compute $g'$

**Logistic function**

$$g(z) = \sigma(z)$$

$$= \frac{1}{1 + e^{-z}}$$
Now, the only thing we need to figure out is how to compute $g'$

**Logistic function**

$$g(z) = \sigma(z)$$

$$= \frac{1}{1 + e^{-z}}$$

$$g'(z) = (-1) \frac{1}{(1 + e^{-z})^2} \frac{d}{dz}(1 + e^{-z})$$
Now, the only thing we need to figure out is how to compute $g'$

**Logistic function**

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = (-1) \frac{1}{(1 + e^{-z})^2} \frac{d}{dz} (1 + e^{-z})$$

$$= (-1) \frac{1}{(1 + e^{-z})^2} (-e^{-z})$$
Now, the only thing we need to figure out is how to compute $g'$. 

**Logistic function**

\[
g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}
\]

\[
g'(z) = (-1) \frac{1}{(1 + e^{-z})^2} \frac{d}{dz} (1 + e^{-z})
\]

\[
= (-1) \frac{1}{(1 + e^{-z})^2} (-e^{-z})
\]

\[
= \frac{1}{1 + e^{-z}} \left( \frac{1 + e^{-z} - 1}{1 + e^{-z}} \right)
\]
Now, the only thing we need to figure out is how to compute $g'$

**Logistic function**

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = (-1) \frac{1}{(1 + e^{-z})^2} \frac{d}{dz} (1 + e^{-z})$$

$$= (-1) \frac{1}{(1 + e^{-z})^2} (-e^{-z})$$

$$= \frac{1}{1 + e^{-z}} \left( \frac{1 + e^{-z} - 1}{1 + e^{-z}} \right)$$

$$= g(z)(1 - g(z))$$
Now, the only thing we need to figure out is how to compute \( g' \)

### Logistic function

\[
g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}
\]

\[
g'(z) = (-1) \frac{1}{(1 + e^{-z})^2} \frac{d}{dz} (1 + e^{-z}) = (1 + e^{-z})^2 (e^{-z} - e^z) = g(z)(1 - g(z))
\]

### tanh

\[
tanh(g(z)) = \frac{e^z - e^{-z}}{e^z + e^{-z}}
\]
Now, the only thing we need to figure out is how to compute $g'$

**Logistic function**

\[
g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}
\]

\[
g'(z) = (-1) \frac{1}{(1 + e^{-z})^2} \frac{d}{dz} (1 + e^{-z})
\]

\[
= (-1) \frac{1}{(1 + e^{-z})^2} (-e^{-z})
\]

\[
= \frac{1}{1 + e^{-z}} \left( \frac{1 + e^{-z} - 1}{1 + e^{-z}} \right)
\]

\[
= g(z)(1 - g(z))
\]

**tanh**

\[
g(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}
\]

\[
g'(z) = \left( (e^z + e^{-z}) \frac{d}{dz} (e^z - e^{-z}) - (e^z - e^{-z}) \frac{d}{dz} (e^z + e^{-z}) \right) \frac{1}{(e^z + e^{-z})^2}
\]

\[
= \frac{1}{1 + e^{-z}} \left( \frac{e^z - e^{-z}}{1 + e^{-z}} \right)
\]
Now, the only thing we need to figure out is how to compute $g'$

**Logistic function**

\[ g(z) = \sigma(z) \]
\[ = \frac{1}{1 + e^{-z}} \]
\[ g'(z) = (-1) \frac{1}{(1 + e^{-z})^2} \frac{d}{dz}(1 + e^{-z}) \]
\[ = (1 + e^{-z}) (-e^{-z}) \]
\[ = \frac{1}{1 + e^{-z}} \left( \frac{1 + e^{-z} - 1}{1 + e^{-z}} \right) \]
\[ = (1 - g(z)) g(z) \]

**tanh**

\[ g(z) = \tanh(z) \]
\[ = \frac{e^z - e^{-z}}{e^z + e^{-z}} \]
\[ g'(z) = \left( \frac{(e^z + e^{-z}) \frac{d}{dz}(e^z - e^{-z})}{(e^z + e^{-z})^2} - (e^z - e^{-z}) \frac{d}{dz}(e^z + e^{-z}) \right) \]
\[ = \frac{(e^z + e^{-z})^2 - (e^z - e^{-z})^2}{(e^z + e^{-z})^2} \]
Now, the only thing we need to figure out is how to compute $g'$.

**Logistic function**

\[
g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}
\]

\[
g'(z) = (1 + e^{-z})^2 \frac{d}{dz}(1 + e^{-z}) = (1 + e^{-z})(1 - g(z))
\]

**tanh**

\[
g(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}
\]

\[
g'(z) = \left(\frac{(e^z + e^{-z}) \frac{d}{dz}(e^z - e^{-z}) - (e^z - e^{-z}) \frac{d}{dz}(e^z + e^{-z})}{e^z + e^{-z}}\right) = \frac{(e^z + e^{-z})^2 - (e^z - e^{-z})^2}{(e^z + e^{-z})^2} = 1 - \frac{(e^z - e^{-z})^2}{(e^z + e^{-z})^2}
\]
Now, the only thing we need to figure out is how to compute \( g' \)

**Logistic function**

\[
g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}
\]

\[
g'(z) = (1) \frac{1}{(1 + e^{-z})^2} \frac{d}{dz} (1 + e^{-z})
\]

\[
= (1) \frac{1}{(1 + e^{-z})^2} (-e^{-z})
\]

\[
= \frac{1}{1 + e^{-z}} \left( \frac{1 + e^{-z} - 1}{1 + e^{-z}} \right)
\]

\[
= g(z)(1 - g(z))
\]

**tanh**

\[
g(z) = \tanh (z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}
\]

\[
g'(z) = \frac{(e^z + e^{-z}) \frac{d}{dz} (e^z - e^{-z}) - (e^z - e^{-z}) \frac{d}{dz} (e^z + e^{-z})}{(e^z + e^{-z})^2}
\]

\[
= \frac{(e^z + e^{-z})^2 - (e^z - e^{-z})^2}{(e^z + e^{-z})^2}
\]

\[
= 1 - \frac{(e^z - e^{-z})^2}{(e^z + e^{-z})^2}
\]

\[
= 1 - (g(z))^2
\]