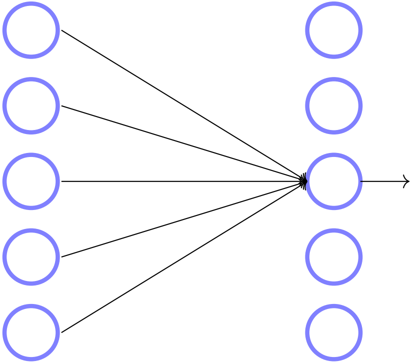
Deep Learning

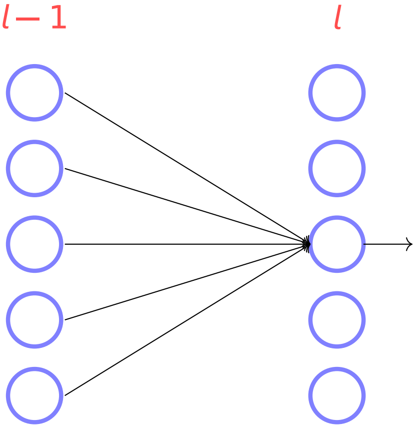
Lecture 7: Backpropagation

Dr. Mehrdad Maleki

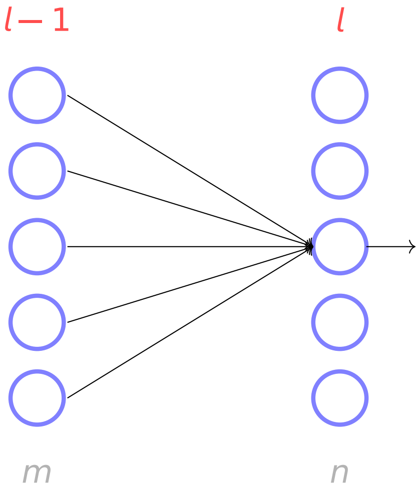
Notations



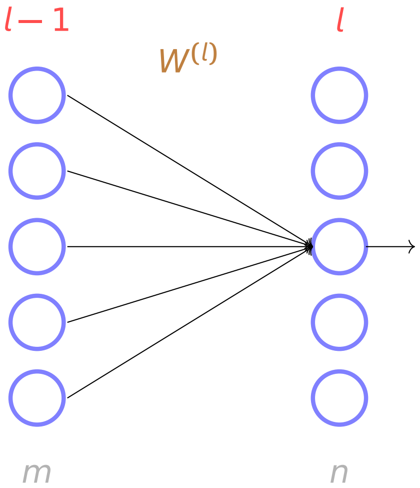
Notations



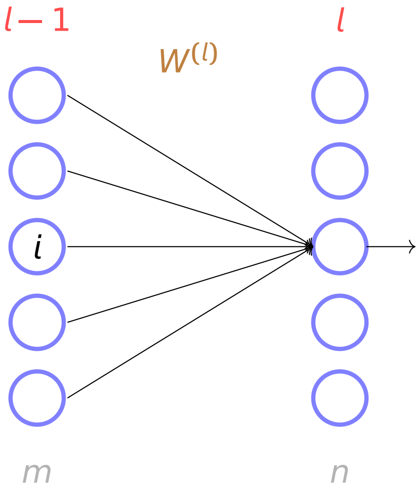
Notations



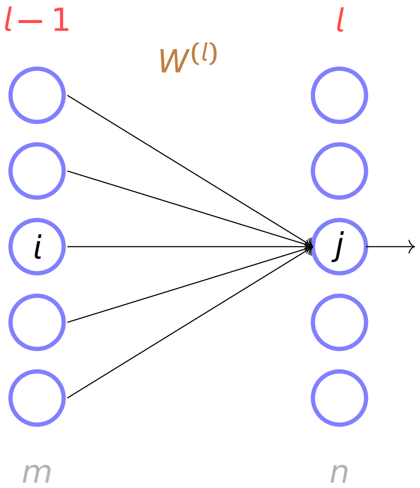
Notations



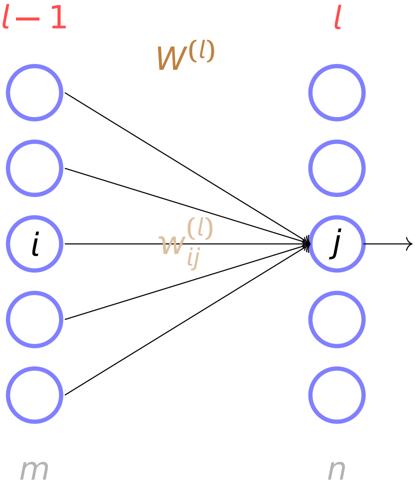
Notations



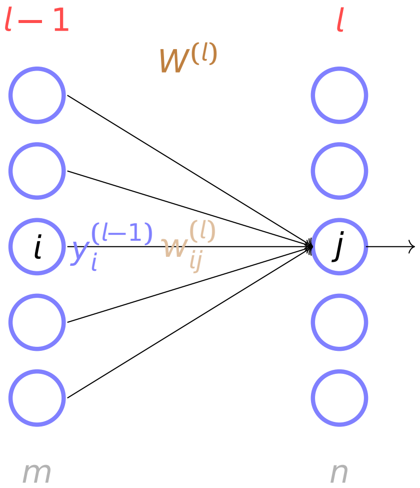
Notations



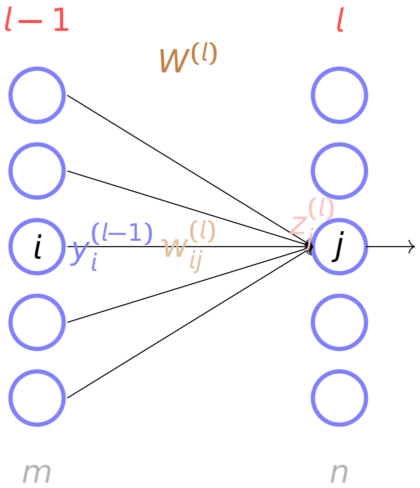
Notations



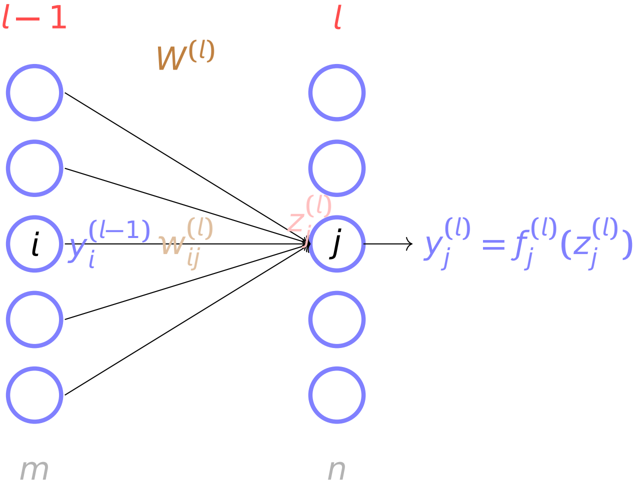
Notations



Notations



Notations



- $z_j^{(l)} = \sum_{k=1}^{m} y_k^{(l-1)} w_{kj}^{(l)} + b_j^{(l)}$

- $z_j^{(l)} = \sum_{k=1}^{m} y_k^{(l-1)} w_{kj}^{(l)} + b_j^{(l)}$
- $y_j^{(l)} = f_j^{(l)}(z_j^{(l)})$

- $z_j^{(l)} = \sum_{k=1}^{m} y_k^{(l-1)} w_{kj}^{(l)} + b_j^{(l)}$
- $y_j^{(l)} = f_j^{(l)}(z_j^{(l)})$
- $y_j^{(0)} = x_j$ the j-th input.

- $z_j^{(l)} = \sum_{k=1}^{m} y_k^{(l-1)} w_{kj}^{(l)} + b_j^{(l)}$
- $y_j^{(l)} = f_j^{(l)}(z_j^{(l)})$
- $y_j^{(0)} = x_j$ the j-th input.
- $y_j^{(L)} =$ the j-th output.

Matrix form

- For $l = 1, \ldots, L$,

$$\mathbf{z}^{(l)} = \begin{bmatrix} z_1^{(l)} \\ \vdots \\ z_{m_l}^{(l)} \end{bmatrix}$$

Matrix form

- For $l = 1, \ldots, L$,

$$\mathbf{z}^{(l)} = \begin{bmatrix} z_1^{(l)} \\ \vdots \\ z_{m_l}^{(l)} \end{bmatrix}$$

$$\mathbf{b}^{(l)} = \begin{bmatrix} b_1^{(l)} \\ \vdots \\ b_{m_l}^{(l)} \end{bmatrix}$$

Matrix form

- For $l = 1, \ldots, L$,

$$\mathbf{z}^{(l)} = \begin{bmatrix} z_1^{(l)} \\ \vdots \\ z_{m_l}^{(l)} \end{bmatrix}$$

$$\mathbf{b}^{(l)} = \begin{bmatrix} b_1^{(l)} \\ \vdots \\ b_{m_l}^{(l)} \end{bmatrix}$$

- For $l = 0, \ldots, L$,

$$\mathbf{y}^{(l)} = \begin{bmatrix} y_1^{(l)} \\ \vdots \\ y_{m_l}^{(l)} \end{bmatrix}$$

where m_l is the number of neurons in layer l.

The weight matrix between layer $l-1$ and l which is a $m_{l-1} \times m_l$ matrix as follow,

$$\mathbf{W}^{(l)} = \begin{bmatrix} w_{11}^{(l)} & \dots & w_{1m_l}^{(l)} \\ \vdots & \ddots & \vdots \\ w_{m_{l-1}1}^{(l)} & \dots & w_{m_{l-1}m_l}^{(l)} \end{bmatrix}$$

- $\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}$

- $\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}$
- $\mathbf{y}^{(l)} = \mathbf{f}^{(l)}(\mathbf{z}^{(l)})$

where,

- $\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}$
- $\mathbf{y}^{(l)} = \mathbf{f}^{(l)}(\mathbf{z}^{(l)})$

where,

$$\mathbf{f}^{(l)} = \begin{bmatrix} f_1^{(l)} \\ \vdots \\ f_{m_l}^{(l)} \end{bmatrix}$$

and

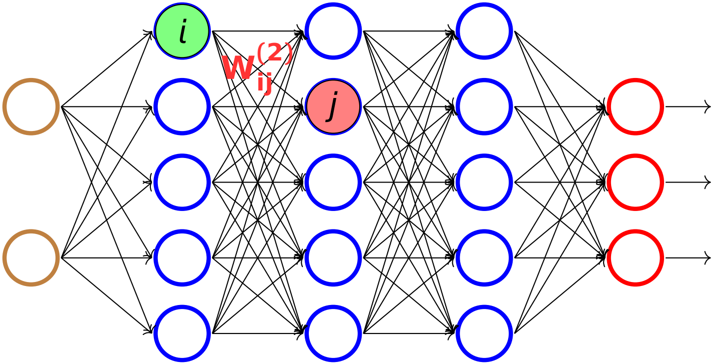
- $\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}$
- $\mathbf{y}^{(l)} = \mathbf{f}^{(l)}(\mathbf{z}^{(l)})$

where,

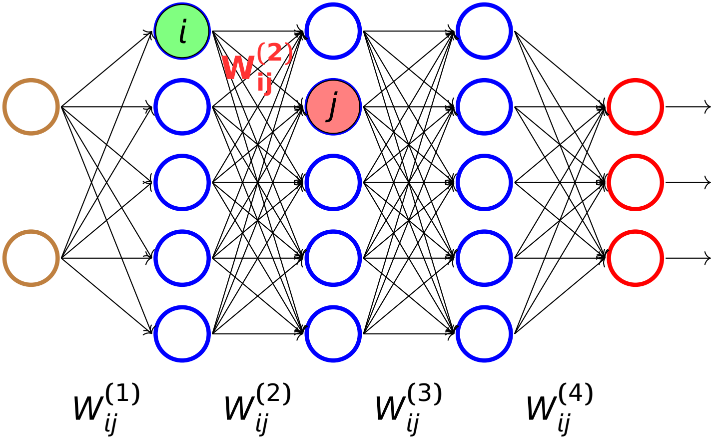
$$\mathbf{f}^{(l)} = \begin{bmatrix} f_1^{(l)} \\ \vdots \\ f_{m_l}^{(l)} \end{bmatrix}$$

and

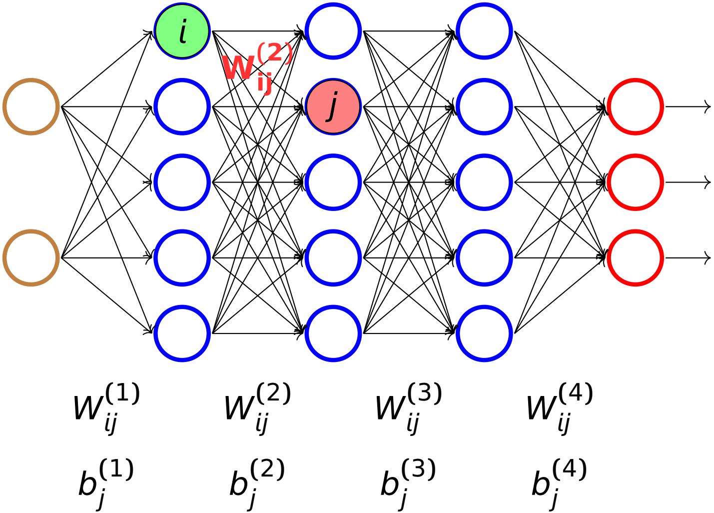
$$\mathbf{f}^{(l)}(\mathbf{z}^{(l)}) = \begin{bmatrix} f_1^{(l)}(z_1^{(l)}) \\ \vdots \\ f_{m_l}^{(l)}(z_{m_l}^{(l)}) \end{bmatrix}$$



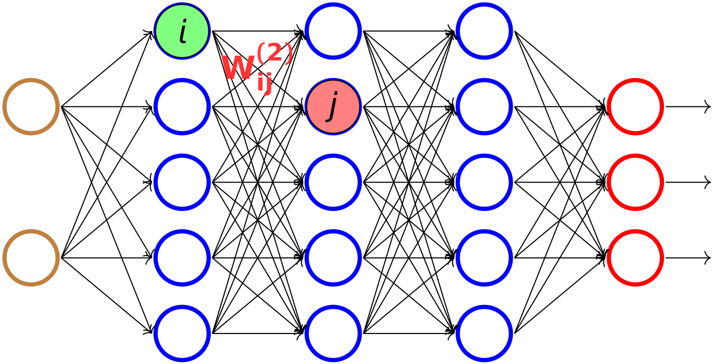
i
$W_{ij}^{(2)}$
j



i
$W_{ij}^{(2)}$
j
$W_{ij}^{(1)}$
$W_{ij}^{(2)}$
$W_{ij}^{(3)}$
$W_{ij}^{(4)}$



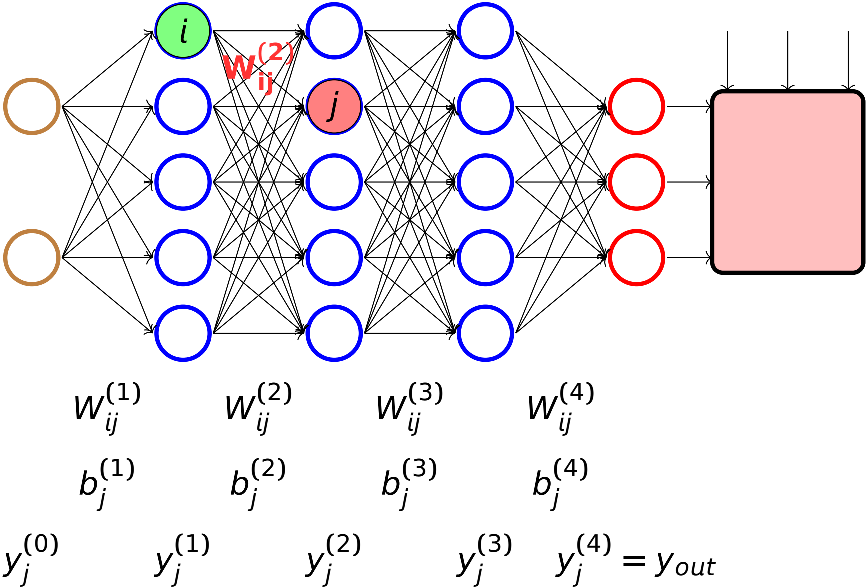
i
$\mathbf{W_{ij}^{(2)}}$
j
$W_{ij}^{(1)}$ $W_{ij}^{(2)}$ $W_{ij}^{(3)}$ $W_{ij}^{(4)}$
$b_j^{(1)}$ $b_j^{(2)}$ $b_j^{(3)}$ $b_j^{(4)}$



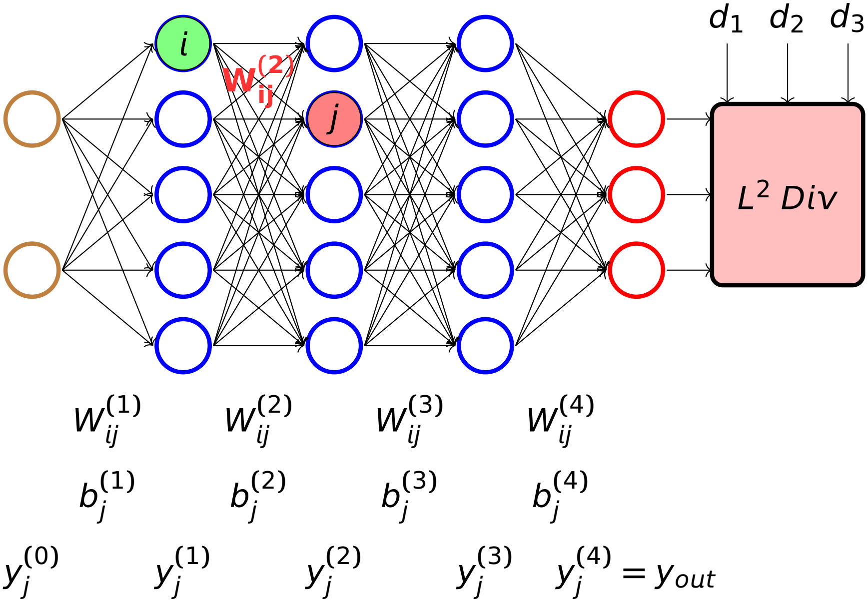
$W_{ij}^{(1)}$ $\quad W_{ij}^{(2)}$ $\quad W_{ij}^{(3)}$ $\quad W_{ij}^{(4)}$

$b_j^{(1)}$ $\quad b_j^{(2)}$ $\quad b_j^{(3)}$ $\quad b_j^{(4)}$

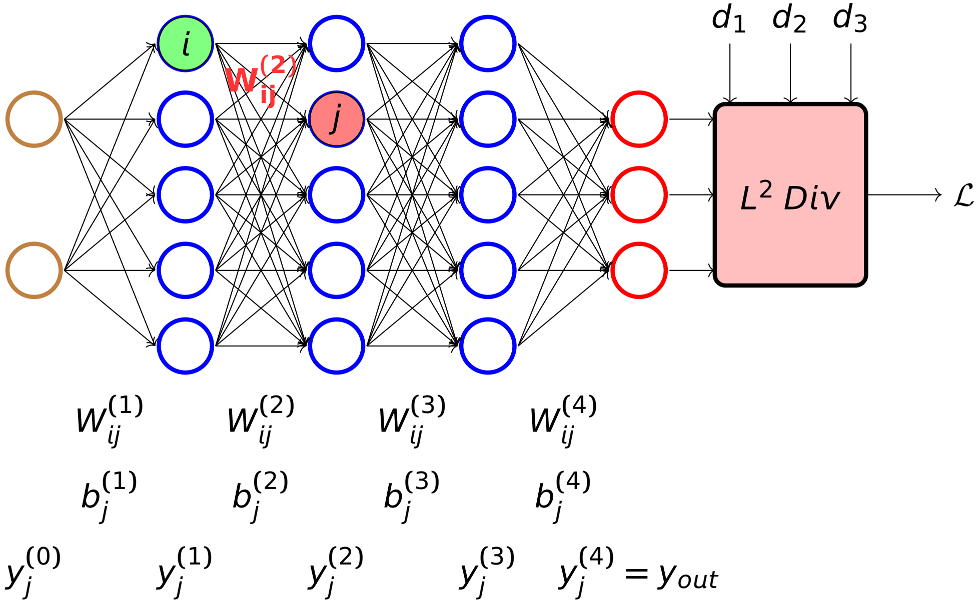
$y_j^{(0)}$ $\quad y_j^{(1)}$ $\quad y_j^{(2)}$ $\quad y_j^{(3)}$ $\quad y_j^{(4)} = y_{out}$



i
j
$\mathbf{W_{ij}^{(2)}}$
$W_{ij}^{(1)}$ $W_{ij}^{(2)}$ $W_{ij}^{(3)}$ $W_{ij}^{(4)}$
$b_j^{(1)}$ $b_j^{(2)}$ $b_j^{(3)}$ $b_j^{(4)}$
$y_j^{(0)}$ $y_j^{(1)}$ $y_j^{(2)}$ $y_j^{(3)}$ $y_j^{(4)} = y_{out}$



d_1 d_2 d_3
i
$\mathbf{w_{ij}^{(2)}}$
j
$L^2\ Div$
$W_{ij}^{(1)}$ $W_{ij}^{(2)}$ $W_{ij}^{(3)}$ $W_{ij}^{(4)}$
$b_j^{(1)}$ $b_j^{(2)}$ $b_j^{(3)}$ $b_j^{(4)}$
$y_j^{(0)}$ $y_j^{(1)}$ $y_j^{(2)}$ $y_j^{(3)}$ $y_j^{(4)} = y_{out}$



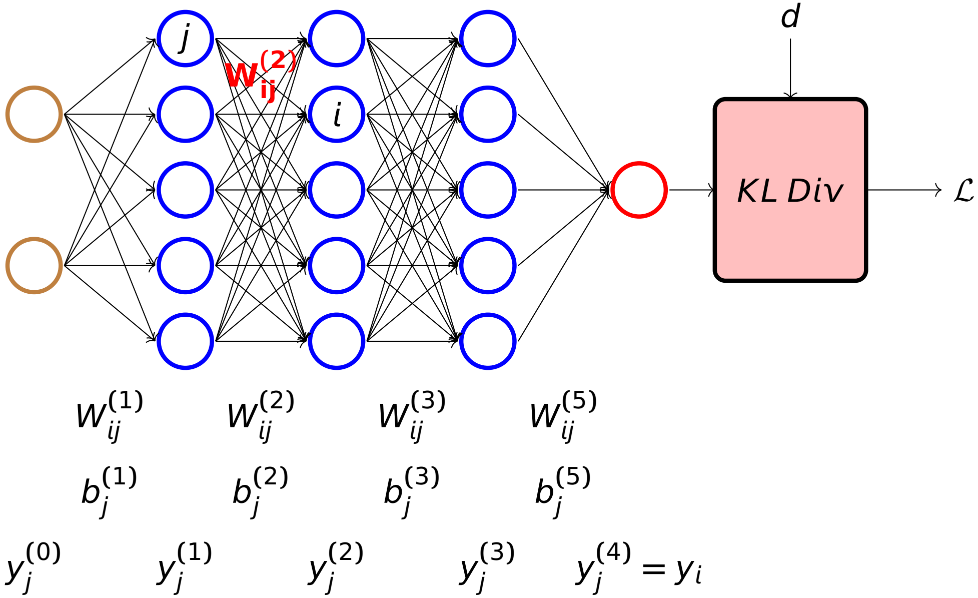
i
j
$W_{ij}^{(2)}$
d_1 d_2 d_3
L^2 Div
$\mathcal{L}$
$W_{ij}^{(1)}$ $W_{ij}^{(2)}$ $W_{ij}^{(3)}$ $W_{ij}^{(4)}$
$b_j^{(1)}$ $b_j^{(2)}$ $b_j^{(3)}$ $b_j^{(4)}$
$y_j^{(0)}$ $y_j^{(1)}$ $y_j^{(2)}$ $y_j^{(3)}$ $y_j^{(4)} = y_{out}$

For real-valued output vector, we use L^2 norm for divergence which is

$$Div(\mathbf{Y}, \mathbf{d}) = \|\mathbf{Y} - \mathbf{d}\|$$

For binary classifier with scalar output, $Y \in (0, 1)$, the desire output is 0 or 1 and the cross entropy between the probability distribution $[Y, 1 - Y]$ and the ideal output probability $[d, 1 - d]$ is the **Kullback–Leibler** divergence, i.e.,

$$Div(Y, d) = -d \log Y - (1 - d) \log(1 - Y)$$


j
W_ij^(2)
i
d
KL Div
$\mathcal{L}$
$W_{ij}^{(1)}$ $W_{ij}^{(2)}$ $W_{ij}^{(3)}$ $W_{ij}^{(5)}$
$b_j^{(1)}$ $b_j^{(2)}$ $b_j^{(3)}$ $b_j^{(5)}$
$y_j^{(0)}$ $y_j^{(1)}$ $y_j^{(2)}$ $y_j^{(3)}$ $y_j^{(4)} = y_i$

Backpropagation

The goal of this section is to calculate $\frac{\partial \mathcal{L}}{\partial w_{ij}^{(1)}}$ recursively.

$$\frac{\partial \mathcal{L}}{\partial y_j^{(3)}} = \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial y_j^{(4)}} \frac{\partial y_j^{(4)}}{\partial z_j^{(4)}} \frac{\partial z_j^{(4)}}{\partial y_j^{(3)}}$$

If we let $\delta y_j^{(l)} = \frac{\partial y}{\partial y_j^{(l)}}$ then,

$$\frac{\partial \mathcal{L}}{\partial y_j^{(1)}} = \frac{\partial \mathcal{L}}{\partial y} \delta y_j^{(1)}$$

Backpropagation

On the other hand

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(1)}} = \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial y_j^{(4)}} \frac{\partial y_j^{(4)}}{\partial z_j^{(4)}} \frac{\partial z_j^{(4)}}{\partial y_j^{(3)}}$$

Backpropagation

On the other hand

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(1)}} = \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial y_j^{(4)}} \frac{\partial y_j^{(4)}}{\partial z_j^{(4)}} \frac{\partial z_j^{(4)}}{\partial y_j^{(3)}} \frac{\partial y_j^{(3)}}{\partial z_j^{(3)}} \frac{\partial z_j^{(3)}}{\partial y_j^{(2)}}$$

Backpropagation

On the other hand

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(1)}} = \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial y_j^{(4)}} \frac{\partial y_j^{(4)}}{\partial z_j^{(4)}} \frac{\partial z_j^{(4)}}{\partial y_j^{(3)}} \\ \frac{\partial y_j^{(3)}}{\partial z_j^{(3)}} \frac{\partial z_j^{(3)}}{\partial y_j^{(2)}} \\ \frac{\partial y_j^{(2)}}{\partial z_j^{(2)}} \frac{\partial z_j^{(2)}}{\partial y_j^{(1)}}$$

Backpropagation

On the other hand

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_{ij}^{(1)}} = \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial y_j^{(4)}} & \frac{\partial y_j^{(4)}}{\partial z_j^{(4)}} \frac{\partial z_j^{(4)}}{\partial y_j^{(3)}} \\
& \frac{\partial y_j^{(3)}}{\partial z_j^{(3)}} \frac{\partial z_j^{(3)}}{\partial y_j^{(2)}} \\
& \frac{\partial y_j^{(2)}}{\partial z_j^{(2)}} \frac{\partial z_j^{(2)}}{\partial y_j^{(1)}} \\
& \frac{\partial y_j^{(1)}}{\partial z_j^{(1)}} \frac{\partial z_j^{(1)}}{\partial w_{ij}^{(1)}}
\end{aligned}
$$

- $\textbf{loss} = \mathcal{L}(\mathbf{y}^{(L)}, \mathbf{d}; \mathbf{W})$

- loss $= \mathcal{L}(\mathbf{y}^{(L)}, \mathbf{d}; \mathbf{W})$
- $\mathbf{y}^{(L)} = \mathbf{f}^{(L)}(\mathbf{z}^{(L)})$

- $\text{loss} = \mathcal{L}(\mathbf{y}^{(L)}, \mathbf{d}; \mathbf{W})$
- $\mathbf{y}^{(L)} = \mathbf{f}^{(L)}(\mathbf{z}^{(L)})$
- $\mathbf{z}^{(L)} = (\mathbf{W}^{(L)})^T \mathbf{y}^{(L-1)} + \mathbf{b}^{(L)}$

- $\text{loss} = \mathcal{L}(\mathbf{y}^{(L)}, \mathbf{d}; \mathbf{W})$
- $\mathbf{y}^{(L)} = \mathbf{f}^{(L)}(\mathbf{z}^{(L)})$
- $\mathbf{z}^{(L)} = (\mathbf{W}^{(L)})^T \mathbf{y}^{(L-1)} + \mathbf{b}^{(L)}$
- $\vdots$
- $\mathbf{y}^{(l)} = \mathbf{f}^{(l)}(\mathbf{z}^{(l)})$

- $\textbf{loss} = \mathcal{L}(\mathbf{y}^{(L)}, \mathbf{d}; \mathbf{W})$
- $\mathbf{y}^{(L)} = \mathbf{f}^{(L)}(\mathbf{z}^{(L)})$
- $\mathbf{z}^{(L)} = (\mathbf{W}^{(L)})^T \mathbf{y}^{(L-1)} + \mathbf{b}^{(L)}$
- $\vdots$
- $\mathbf{y}^{(l)} = \mathbf{f}^{(l)}(\mathbf{z}^{(l)})$
- $\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}$

- $\text{loss} = \mathcal{L}(\mathbf{y}^{(L)}, \mathbf{d}; \mathbf{W})$
- $\mathbf{y}^{(L)} = \mathbf{f}^{(L)}(\mathbf{z}^{(L)})$
- $\mathbf{z}^{(L)} = (\mathbf{W}^{(L)})^T \mathbf{y}^{(L-1)} + \mathbf{b}^{(L)}$
- $\vdots$
- $\mathbf{y}^{(l)} = \mathbf{f}^{(l)}(\mathbf{z}^{(l)})$
- $\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}$
- $\vdots$
- $\mathbf{y}^{(1)} = \mathbf{f}^{(1)}(\mathbf{z}^{(1)})$

- $\text{loss} = \mathcal{L}(\mathbf{y}^{(L)}, \mathbf{d}; \mathbf{W})$
- $\mathbf{y}^{(L)} = \mathbf{f}^{(L)}(\mathbf{z}^{(L)})$
- $\mathbf{z}^{(L)} = (\mathbf{W}^{(L)})^T \mathbf{y}^{(L-1)} + \mathbf{b}^{(L)}$
- $\vdots$
- $\mathbf{y}^{(l)} = \mathbf{f}^{(l)}(\mathbf{z}^{(l)})$
- $\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}$
- $\vdots$
- $\mathbf{y}^{(1)} = \mathbf{f}^{(1)}(\mathbf{z}^{(1)})$
- $\mathbf{z}^{(1)} = (\mathbf{W}^{(1)})^T \mathbf{y}^{(0)} + \mathbf{b}^{(1)}$

- $\text{loss} = \mathcal{L}(\mathbf{y}^{(L)}, \mathbf{d}; \mathbf{W})$
- $\mathbf{y}^{(L)} = \mathbf{f}^{(L)}(\mathbf{z}^{(L)})$
- $\mathbf{z}^{(L)} = (\mathbf{W}^{(L)})^T \mathbf{y}^{(L-1)} + \mathbf{b}^{(L)}$
- $\vdots$
- $\mathbf{y}^{(l)} = \mathbf{f}^{(l)}(\mathbf{z}^{(l)})$
- $\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}$
- $\vdots$
- $\mathbf{y}^{(1)} = \mathbf{f}^{(1)}(\mathbf{z}^{(1)})$
- $\mathbf{z}^{(1)} = (\mathbf{W}^{(1)})^T \mathbf{y}^{(0)} + \mathbf{b}^{(1)}$

We want to compute $\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}}$. But,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{y}^{(L)}} \frac{\partial \mathbf{y}^{(L)}}{\partial \mathbf{y}^{(l)}} \frac{\partial \mathbf{y}^{(l)}}{\partial \mathbf{z}^{(l)}} \frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{W}^{(l)}}$$

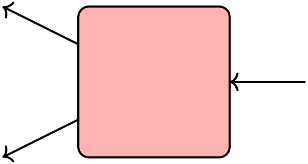
We can compute $\frac{\partial \mathbf{y}^{(L)}}{\partial \mathbf{y}^{(l)}}$ by the chain rule as follow,

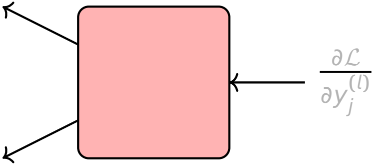
$$\frac{\partial \mathbf{y}^{(L)}}{\partial \mathbf{y}^{(l)}} = \frac{\partial \mathbf{y}^{(L)}}{\partial \mathbf{z}^{(L)}} \frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{y}^{(L-1)}} \frac{\partial \mathbf{y}^{(L-1)}}{\partial \mathbf{y}^{(l)}}$$

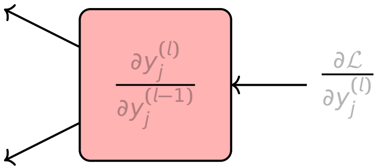
By getting transpose we have

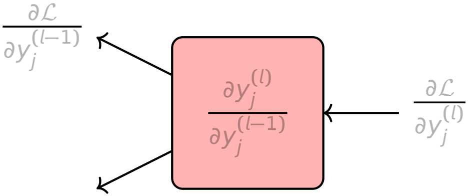
$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}}^T = \frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{W}^{(l)}}^T \frac{\partial \mathbf{y}^{(l)}}{\partial \mathbf{z}^{(l)}}^T \frac{\partial \mathbf{y}^{(L)}}{\partial \mathbf{y}^{(l)}}^T \frac{\partial \mathcal{L}}{\partial \mathbf{y}^{(L)}}^T$$

This is Backpropagation!

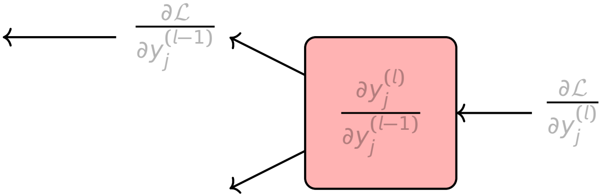



$\frac{\partial \mathcal{L}}{\partial y_j^{(l)}}$

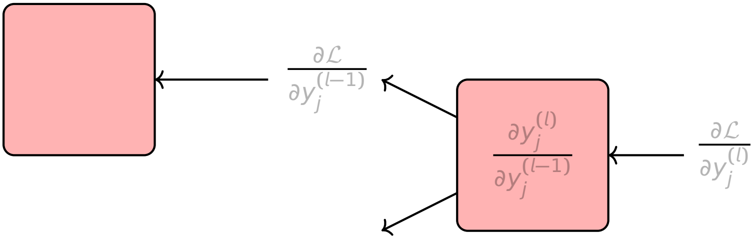

$\frac{\partial y_j^{(l)}}{\partial y_j^{(l-1)}}$
$\frac{\partial \mathcal{L}}{\partial y_j^{(l)}}$

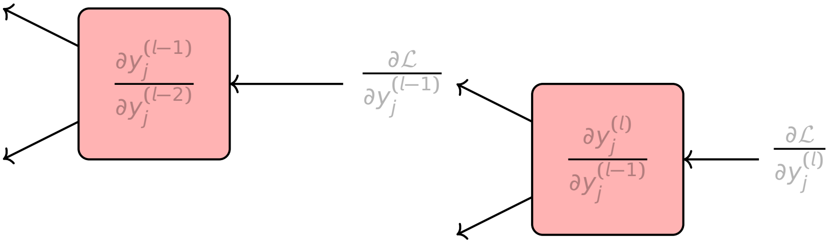


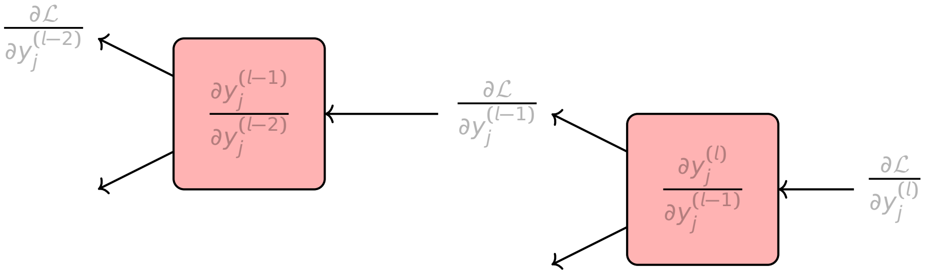
$\frac{\partial \mathcal{L}}{\partial y_j^{(l-1)}}$
$\frac{\partial y_j^{(l)}}{\partial y_j^{(l-1)}}$
$\frac{\partial \mathcal{L}}{\partial y_j^{(l)}}$



$\frac{\partial \mathcal{L}}{\partial y_j^{(l-1)}}$
$\frac{\partial y_j^{(l)}}{\partial y_j^{(l-1)}}$
$\frac{\partial \mathcal{L}}{\partial y_j^{(l)}}$


$\frac{\partial \mathcal{L}}{\partial y_j^{(l-1)}}$
$\frac{\partial y_j^{(l)}}{\partial y_j^{(l-1)}}$
$\frac{\partial \mathcal{L}}{\partial y_j^{(l)}}$


$\frac{\partial y_j^{(l-1)}}{\partial y_j^{(l-2)}}$
$\frac{\partial \mathcal{L}}{\partial y_j^{(l-1)}}$
$\frac{\partial y_j^{(l)}}{\partial y_j^{(l-1)}}$
$\frac{\partial \mathcal{L}}{\partial y_j^{(l)}}$



$\frac{\partial \mathcal{L}}{\partial y_j^{(l-2)}}$
$\frac{\partial y_j^{(l-1)}}{\partial y_j^{(l-2)}}$
$\frac{\partial \mathcal{L}}{\partial y_j^{(l-1)}}$
$\frac{\partial y_j^{(l)}}{\partial y_j^{(l-1)}}$
$\frac{\partial \mathcal{L}}{\partial y_j^{(l)}}$

All types of Gradient Descent

1. **Batch Gradient Descent:** uses entire dataset for one update.

2. **Minibatch Gradient Descent:** uses subsets of the dataset for one update.

3. **Stochastic Gradient Descent (online):** uses single example for one update.

Downsides

1. **Batch Gradient Descent:** sensitivity to saddle points and time-complexity, choosing a proper learning rate can be difficult.

2. **Minibatch Gradient Descent:** does not guarantee good convergence.

3. **Stochastic Gradient Descent (online):** time-complexity, stuck at local minima.

Gradient

Let $\{(\mathbf{X_1}, y_1), \ldots, (\mathbf{X_N}, y_N)\}$ be the training set. then the update rules are as follow,

1. **Batch Gradient Descent:**

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \alpha \nabla_{\mathbf{W}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathbf{X_i}, y_i; \mathbf{W}_k)$$

2. **Minibatch Gradient Descent:**

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \alpha \nabla_{\mathbf{W}} \frac{1}{\textbf{batch size}} \sum_{i \in \textbf{minibatch}} \mathcal{L}(\mathbf{X_i}, y_i; \mathbf{W}_k)$$

3. **Stochastic Gradient Descent (online):**

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \alpha \nabla_{\mathbf{W}} \mathcal{L}(\mathbf{X_i}, y_i; \mathbf{W}_k)$$

Thank You