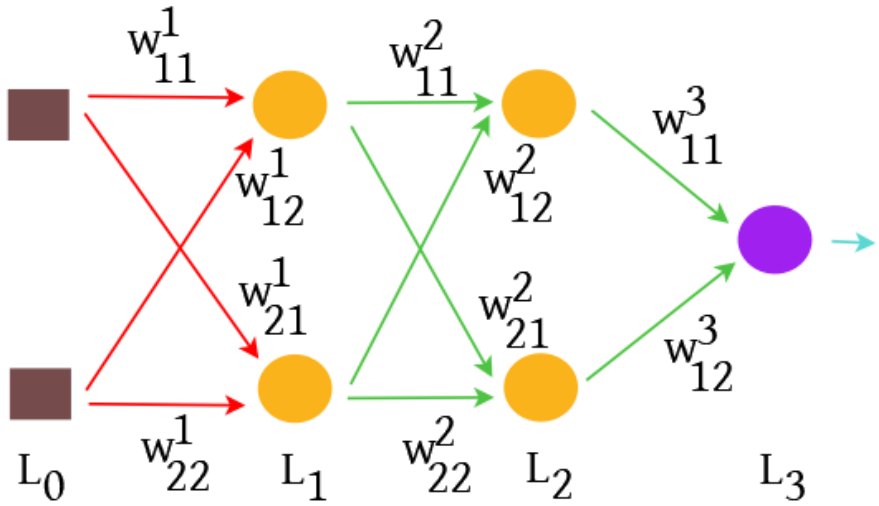
Deep Learning - Theory and Practice

IE 643
Lecture 8

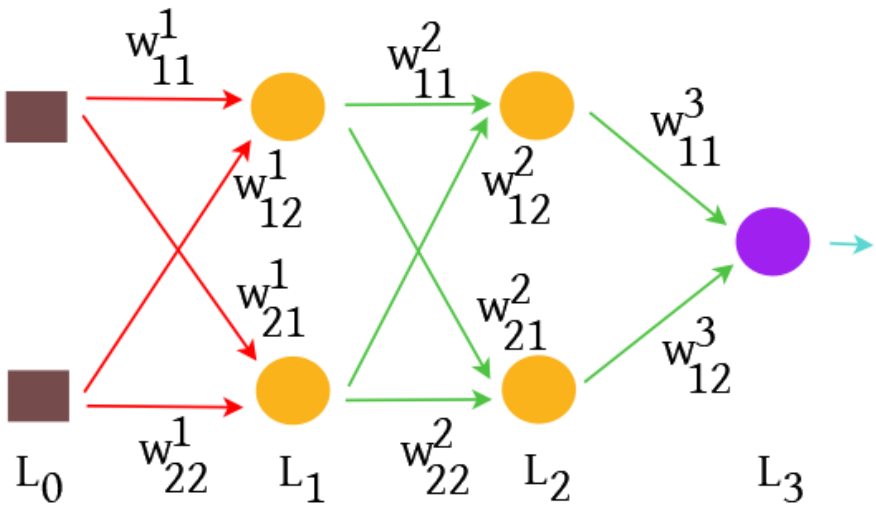
August 27, 2022.

Multi Layer Perceptron - Data Perspective



- **Input:** Training Data $D = \{(x^s, y^s)\}_{s=1}^{S}$.
- For each sample x^s the prediction $\hat{y}^s = \text{MLP}(x^s)$.
- **Error:** $e^s = E(y^s, \hat{y}^s)$.
- **Aim:** To minimize $\sum_{s=1}^{S} e^s$.

Multi Layer Perceptron - Data Perspective

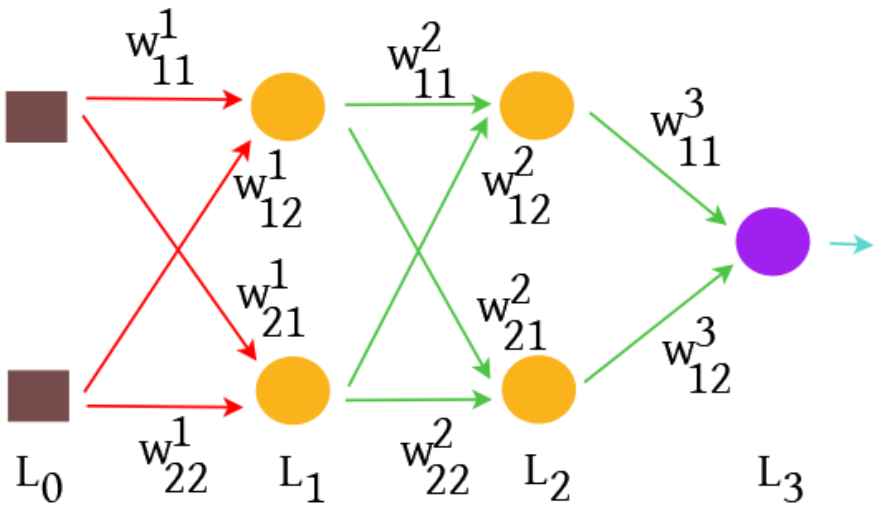


Optimization perspective

- Given training data $D = \{(x^s, y^s)\}_{s=1}^{S}$,

$$\min \sum_{s=1}^{S} e^s$$

Multi Layer Perceptron - Data Perspective

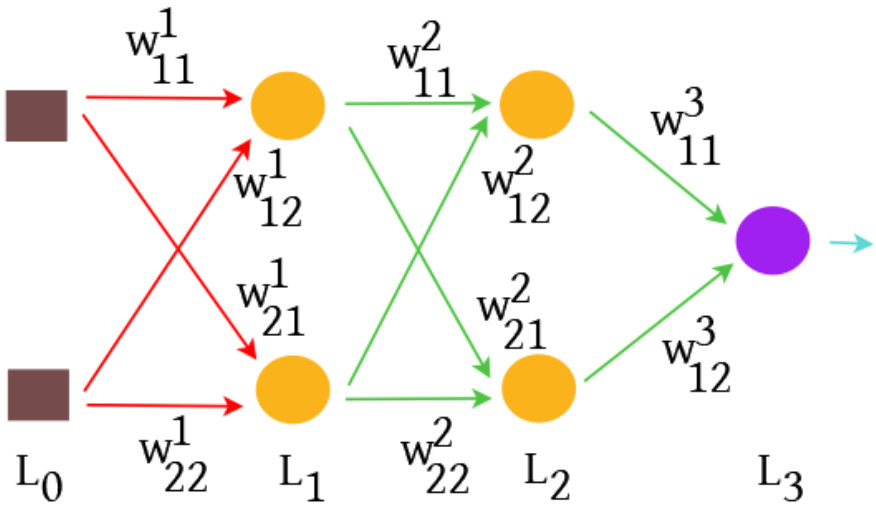


Optimization perspective

- Given training data $D = \{(x^s, y^s)\}_{s=1}^{S}$,

$$\min \sum_{s=1}^{S} e^s = \sum_{s=1}^{S} E(y^s, \hat{y}^s)$$

Multi Layer Perceptron - Data Perspective

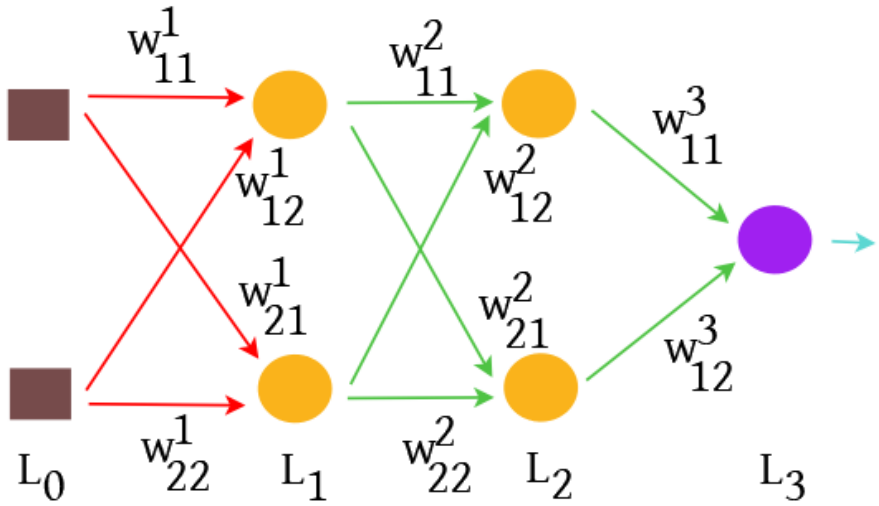


Optimization perspective

- Given training data $D = \{(x^s, y^s)\}_{s=1}^{S}$,

$$\min \sum_{s=1}^{S} e^s = \sum_{s=1}^{S} E(y^s, \hat{y}^s) = \sum_{s=1}^{S} E(y^s, \text{MLP}(x^s))$$

Multi Layer Perceptron - Data Perspective



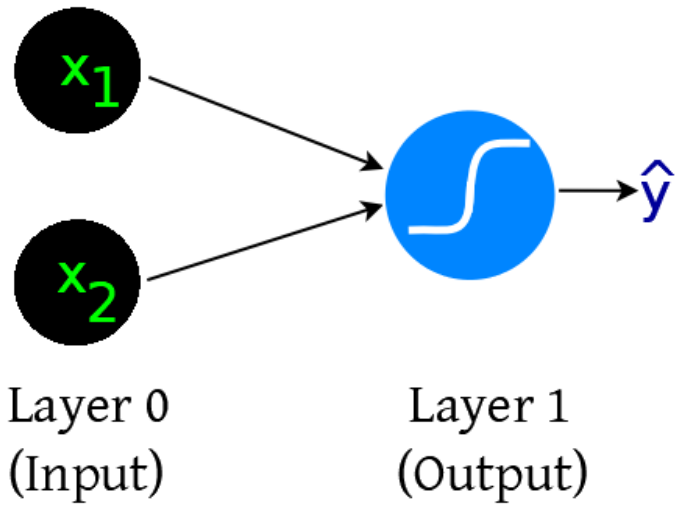
Optimization perspective

- Given training data $D = \{(x^s, y^s)\}_{s=1}^{S}$,

$$\min \sum_{s=1}^{S} e^s = \sum_{s=1}^{S} E(y^s, \hat{y}^s) = \sum_{s=1}^{S} E(y^s, \text{MLP}(x^s))$$

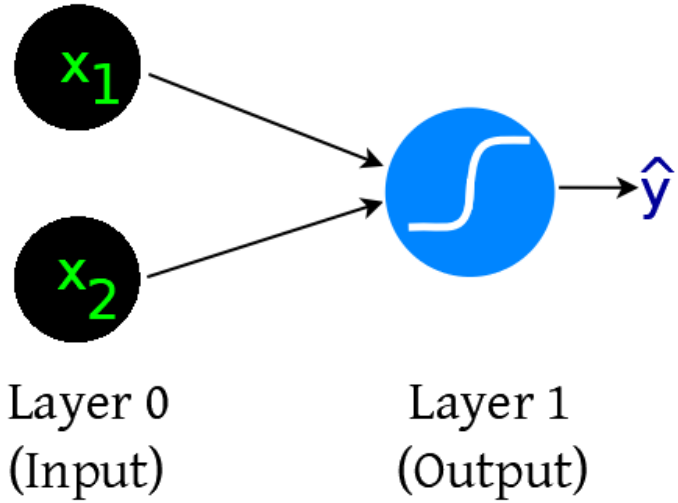
- Note: The minimization is over the weights of the MLP $W^1, \ldots, W^L$, where L denotes number of layers in MLP.

MLP - Data Perspective: A Simple Example



$$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2) = \frac{1}{1 + \exp\left(-[w_{11}^1 x_1 + w_{12}^1 x_2]\right)}$$

MLP - Data Perspective: A Simple Example

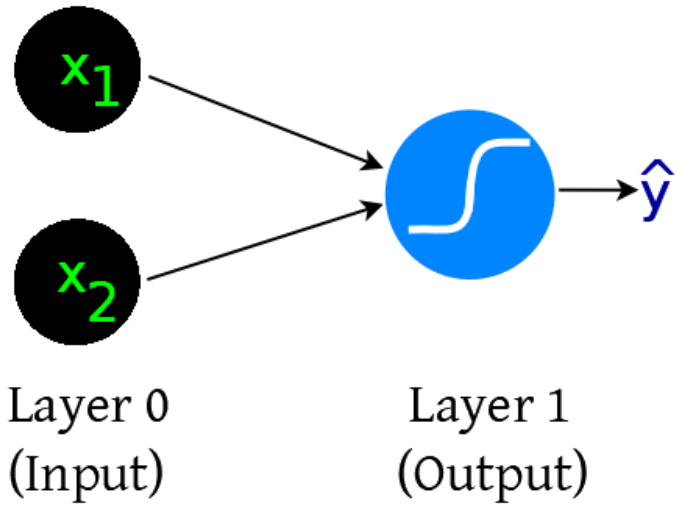


$$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2) = \frac{1}{1 + \exp\left(-[w_{11}^1 x_1 + w_{12}^1 x_2]\right)}$$

Property of 0-1 sigmoid $\sigma : \mathbb{R} \to [0, 1]$

- σ is continuous
- σ is monotonic
- $\sigma(z) \to \begin{cases} 0 & \text{if } z \to -\infty \\ 1 & \text{if } z \to +\infty \end{cases}$

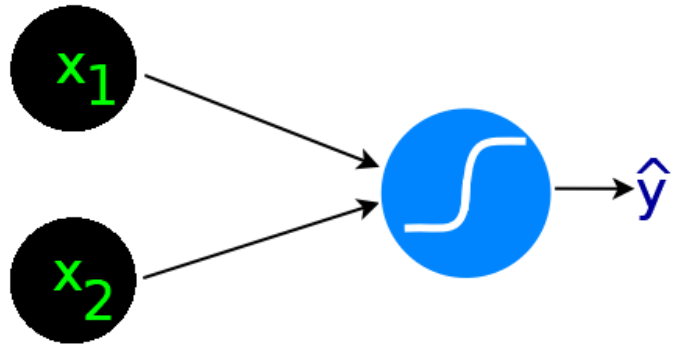
MLP - Data Perspective: A Simple Example



- Let

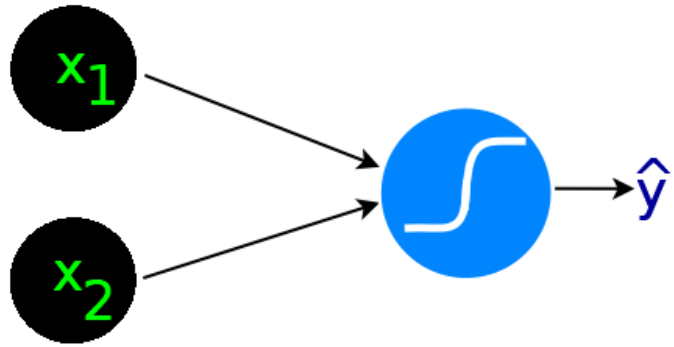
$$D = \{(x^1 = (-3, -3), y^1 = 1), \\ (x^2 = (-2, -2), y^2 = 1), \\ (x^3 = (4, 4), y^3 = 0), \\ (x^4 = (2, -5), y^4 = 0)\}.$$

MLP - Data Perspective: A Simple Example



x_1	x_2	y	$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2)$
-3	-3	1	$\sigma(-3w_{11}^1 - 3w_{12}^1)$
-2	-2	1	$\sigma(-2w_{11}^1 - 2w_{12}^1)$
4	4	0	$\sigma(4w_{11}^1 + 4w_{12}^1)$
2	-5	0	$\sigma(2w_{11}^1 - 5w_{12}^1)$

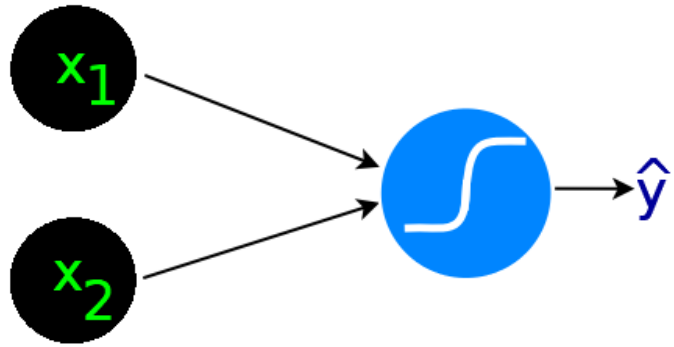
MLP - Data Perspective: A Simple Example



x_1	x_2	y	$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2)$
-3	-3	1	$\sigma(-3w_{11}^1 - 3w_{12}^1)$
-2	-2	1	$\sigma(-2w_{11}^1 - 2w_{12}^1)$
4	4	0	$\sigma(4w_{11}^1 + 4w_{12}^1)$
2	-5	0	$\sigma(2w_{11}^1 - 5w_{12}^1)$

- **Assume:** $\text{Err}(y, \hat{y}) = (y - \hat{y})^2$.

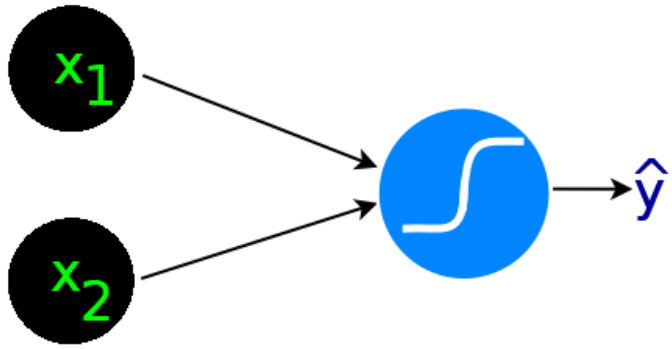
MLP - Data Perspective: A Simple Example



x_1	x_2	y	$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2)$
-3	-3	1	$\sigma(-3w_{11}^1 - 3w_{12}^1)$
-2	-2	1	$\sigma(-2w_{11}^1 - 2w_{12}^1)$
4	4	0	$\sigma(4w_{11}^1 + 4w_{12}^1)$
2	-5	0	$\sigma(2w_{11}^1 - 5w_{12}^1)$

- **Assume:** $\text{Err}(y, \hat{y}) = (y - \hat{y})^2$.
- Popularly called the squared error.

MLP - Data Perspective: A Simple Example



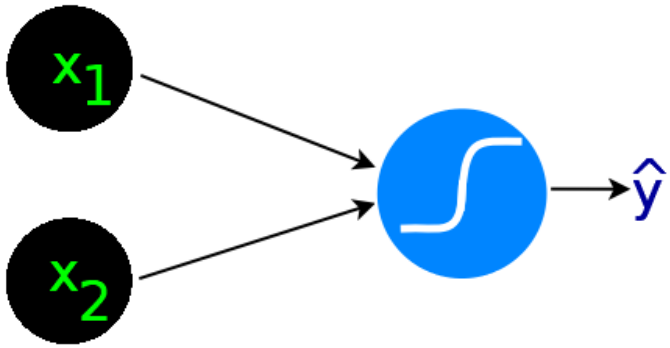
Layer 0 (Input) Layer 1 (Output)

x_1	x_2	y	$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2)$
-3	-3	1	$\sigma(-3w_{11}^1 - 3w_{12}^1)$
-2	-2	1	$\sigma(-2w_{11}^1 - 2w_{12}^1)$
4	4	0	$\sigma(4w_{11}^1 + 4w_{12}^1)$
2	-5	0	$\sigma(2w_{11}^1 - 5w_{12}^1)$

- Total error (or loss):

$$E = \sum_{i=1}^{4} e^i = \sum_{i=1}^{4} \text{Err}(y^i, \hat{y}^i)$$

MLP - Data Perspective: A Simple Example



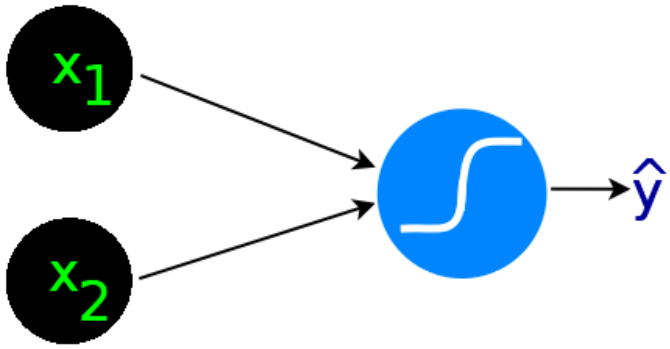
Layer 0 (Input) Layer 1 (Output)

x_1	x_2	y	$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2)$
-3	-3	1	$\sigma(-3w_{11}^1 - 3w_{12}^1)$
-2	-2	1	$\sigma(-2w_{11}^1 - 2w_{12}^1)$
4	4	0	$\sigma(4w_{11}^1 + 4w_{12}^1)$
2	-5	0	$\sigma(2w_{11}^1 - 5w_{12}^1)$

- Total error (or loss):

$$E = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

MLP - Data Perspective: A Simple Example



Layer 0 (Input) Layer 1 (Output)

x_1	x_2	y	$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2)$
-3	-3	1	$\sigma(-3w_{11}^1 - 3w_{12}^1)$
-2	-2	1	$\sigma(-2w_{11}^1 - 2w_{12}^1)$
4	4	0	$\sigma(4w_{11}^1 + 4w_{12}^1)$
2	-5	0	$\sigma(2w_{11}^1 - 5w_{12}^1)$

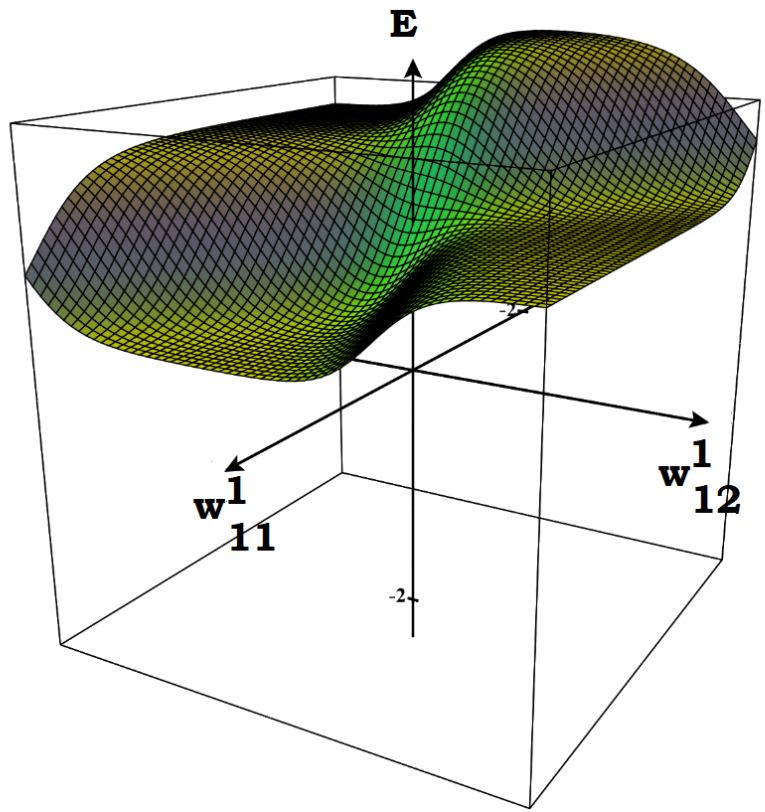
- Aim: To minimize the total error (or loss), which is

$$\min_{w_{11}^1, w_{12}^1} E = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

MLP - Data Perspective: A Simple Example

Visualizing the loss surface:

x_1	x_2	y	$\hat{y} = \sigma(w_{11}^1 x_1 + w_{12}^1 x_2)$
-3	-3	1	$\sigma(-3w_{11}^1 - 3w_{12}^1)$
-2	-2	1	$\sigma(-2w_{11}^1 - 2w_{12}^1)$
4	4	0	$\sigma(4w_{11}^1 + 4w_{12}^1)$
2	-5	0	$\sigma(2w_{11}^1 - 5w_{12}^1)$



$$E = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

Optimization Concepts

General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x)$$

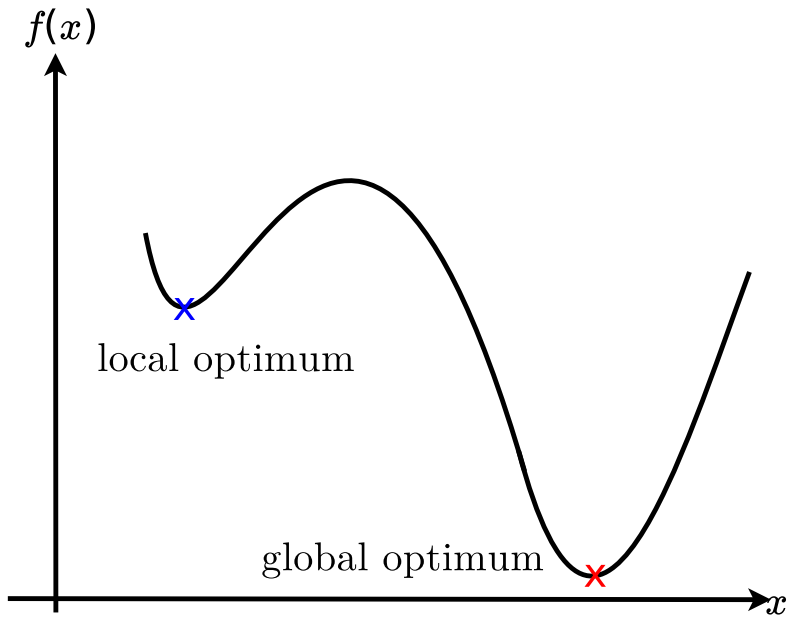
General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x)$$

- f is called objective function and $\mathcal{C}$ is called feasible set.
- Let $f^* = \min_{x \in \mathcal{C}} f(x)$ denote the **optimal objective function value**.
- **Optimal Solution Set** $S^* = \{x \in \mathcal{C} : f(x) = f^*\}$.
- Let us denote by x^* an optimal solution in S^*.

General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x) \qquad \text{(OP)}$$



General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x) \qquad \text{(OP)}$$

Local Optimal Solution

A solution z to (OP) is called local optimal solution if $f(z) \leq f(\hat{z})$, $\forall \hat{z} \in \mathcal{N}(z, \epsilon)$ for some $\epsilon > 0$.

Note: $\mathcal{N}(z, \epsilon)$ denotes suitable $\epsilon-$neighborhood of z.

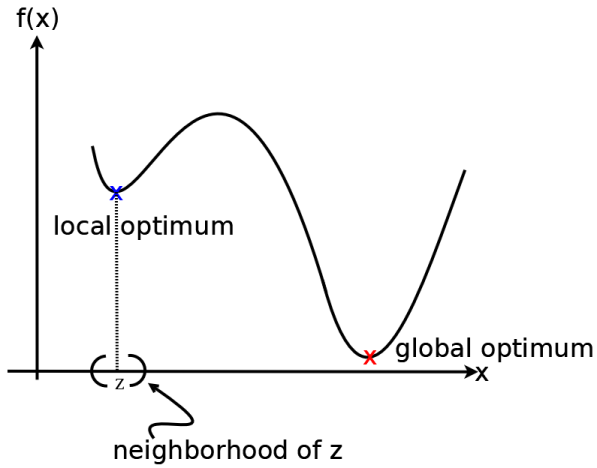
General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x) \qquad \text{(OP)}$$

Local Optimal Solution

A solution z to (OP) is called local optimal solution if $f(z) \leq f(\hat{z})$, $\forall \hat{z} \in \mathcal{N}(z, \epsilon)$ for some $\epsilon > 0$.

Note: $\mathcal{N}(z, \epsilon)$ denotes suitable $\epsilon-$neighborhood of z.



General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x) \qquad \text{(OP)}$$

Local Optimal Solution

A solution z to (OP) is called local optimal solution if $f(z) \leq f(\hat{z})$, $\forall \hat{z} \in \mathcal{N}(z, \epsilon)$ for some $\epsilon > 0$.

Note: $\mathcal{N}(z, \epsilon)$ denotes suitable $\epsilon-$neighborhood of z.

$\epsilon-$ Neighborhood of $z \in \mathcal{C}$

$$\mathcal{N}(z, \epsilon) = \{u \in \mathcal{C} : \text{dist}(z, u) \leq \epsilon\}.$$

General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x) \qquad \text{(OP)}$$

Local Optimal Solution

A solution z to (OP) is called local optimal solution if $f(z) \leq f(\hat{z})$, $\forall \hat{z} \in \mathcal{N}(z, \epsilon)$ for some $\epsilon > 0$.

Global Optimal Solution

A solution z to (OP) is called global optimal solution if $f(z) \leq f(\hat{z})$, $\forall \hat{z} \in \mathcal{C}$.

General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x)$$

- **General Assumption:** $\mathcal{C} \subseteq \mathbb{R}^d$.

High Dimensional Representation - Notations

- Gradient of a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ at a point x

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \vdots \\ \vdots \\ \frac{\partial f(x)}{\partial x_d} \end{pmatrix}$$

General Optimization Problem

$$\min_{x \in \mathcal{C}} f(x)$$

- $\mathcal{C} \subseteq \mathbb{R}^d$.
- $f : \mathcal{C} \longrightarrow \mathbb{R}$.

Directional derivative

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function defined over $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x \in int(\mathcal{C})$. Let $\mathbf{0} \neq d \in \mathbb{R}^d$. If the limit

$$\lim_{\alpha \downarrow 0} \frac{f(x + \alpha d) - f(x)}{\alpha}$$

exists, then it is called the directional derivative of f at x along the direction d, and is denoted by $f'(x; d)$.

Directional derivative

Interior of a set $\mathcal{C}$

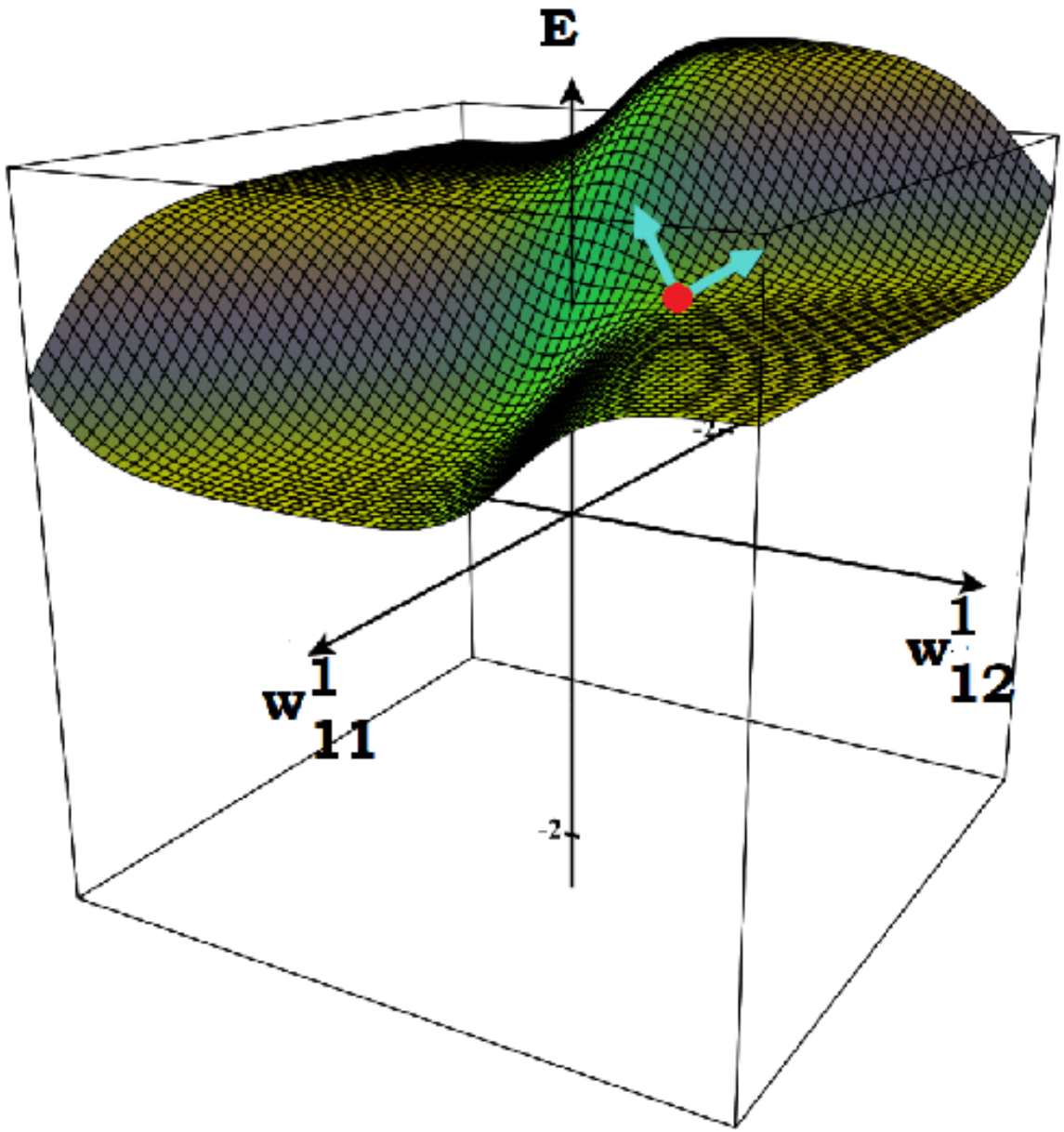
Let $\mathcal{C} \subseteq \mathbb{R}^d$. Then $int(\mathcal{C})$ is defined by:

$$int(\mathcal{C}) = \{x \in \mathcal{C} : B(x, \epsilon) \subseteq \mathcal{C}, \text{ for some } \epsilon > 0\},$$

where $B(x, \epsilon)$ is the open ball centered at x with radius ϵ given by

$$B(x, \epsilon) = \{y \in \mathcal{C} : \|x - y\| < \epsilon\}.$$

Directional derivative



Directional derivative

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function defined over $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x \in int(\mathcal{C})$. Let $d \neq \mathbf{0} \in \mathbb{R}^d$. If the limit

$$\lim_{\alpha \downarrow 0} \frac{f(x + \alpha d) - f(x)}{\alpha}$$

exists, then it is called the directional derivative of f at x along the direction d, and is denoted by $f'(x; d)$.

Note: If all partial derivatives of f exist at x, then $f'(x; d) = \langle \nabla f(x), d \rangle$, where $\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1} \quad \ldots \frac{\partial f(x)}{\partial x_d} \right]^\top$.

Descent Direction

Let $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^d$. Then a vector $\mathbf{0} \neq d \in \mathbb{R}^d$ is called a descent direction of f at x if the directional derivative of f at x is negative; that is,

$$f'(x; d) = \langle \nabla f(x), d \rangle < 0.$$

Descent Direction

Let $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^d$. Then a vector $\mathbf{0} \neq d \in \mathbb{R}^d$ is called a descent direction of f at x if the directional derivative derivative of f at x is negative; that is,

$$f'(x; d) = \langle \nabla f(x), d \rangle < 0.$$

Note: A natural candidate for a descent direction is $d = -\nabla f(x)$.

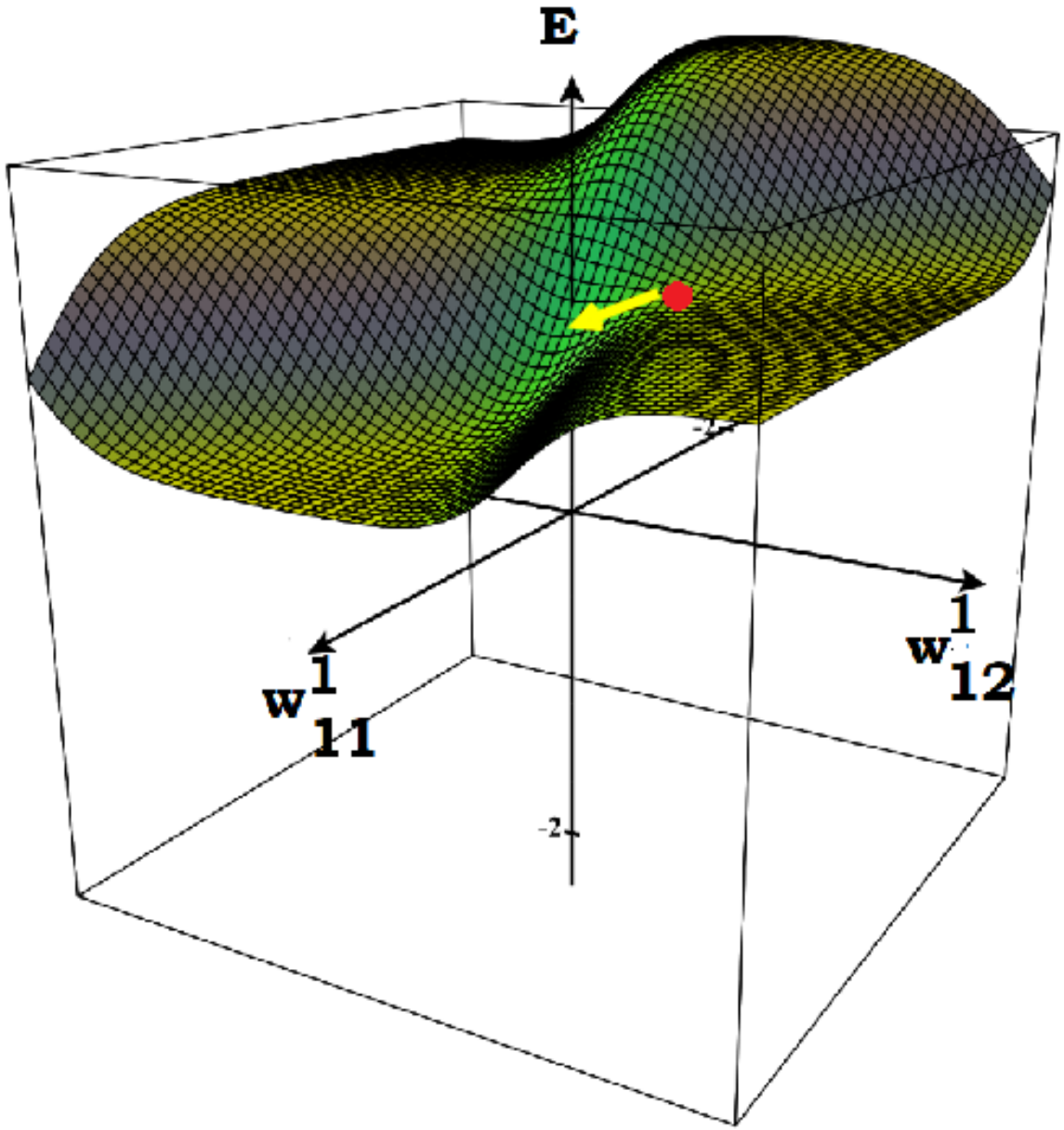
Descent Direction

Proposition

Let $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^d$. Let $\mathbf{0} \neq d \in \mathbb{R}^d$ be a descent direction of f at x. Then there exists $\epsilon > 0$ such that $\forall \alpha \in (0, \epsilon]$ we have

$$f(x + \alpha d) < f(x).$$

Descent Direction



Descent Direction

Proposition

Let $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^d$. Let $\mathbf{0} \neq d \in \mathbb{R}^d$ be a descent direction of f at x. Then there exists $\epsilon > 0$ such that $\forall \alpha \in (0, \epsilon]$ we have

$$f(x + \alpha d) < f(x).$$

Proof idea:

Descent Direction

Proposition

Let $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^d$. Let $\mathbf{0} \neq d \in \mathbb{R}^d$ be a descent direction of f at x. Then there exists $\epsilon > 0$ such that $\forall \alpha \in (0, \epsilon]$ we have

$$f(x + \alpha d) < f(x).$$

Proof idea: Since $\mathbf{0} \neq d \in \mathbb{R}^d$ is a descent direction, by definition of the directional derivative we have

$$f'(x; d) = \lim_{\alpha \downarrow 0} \frac{f(x + \alpha d) - f(x)}{\alpha} < 0$$

$\implies \exists \epsilon > 0$ such that $\forall \alpha \in (0, \epsilon]$, $f(x + \alpha d) < f(x)$.

Descent Direction

Proposition

Let $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^d$. Let $\mathbf{0} \neq d \in \mathbb{R}^d$ be a descent direction of f at x. Then there exists $\epsilon > 0$ such that $\forall \alpha \in (0, \epsilon]$ we have

$$f(x + \alpha d) < f(x).$$

Proof idea: Since $\mathbf{0} \neq d \in \mathbb{R}^d$ is a descent direction, by definition of the directional derivative we have

$$f'(x; d) = \lim_{\alpha \downarrow 0} \frac{f(x + \alpha d) - f(x)}{\alpha} < 0$$

$\implies \exists \epsilon > 0$ such that $\forall \alpha \in (0, \epsilon]$, $f(x + \alpha d) < f(x)$.

Note: If we cannot find such ϵ, d is no longer a descent direction. **Why?**

Algorithm Development using Descent Direction

Consider the general optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \qquad \text{(GEN-OPT)}$$

where $f : \mathbb{R}^d \longrightarrow \mathbb{R}$

Algorithm to solve (GEN-OPT)

- Start with $x^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - Find a descent direction d^k of f at x^k and $\alpha^k > 0$ such that $f(x^k + \alpha^k d^k) < f(x^k)$.
  - $x^{k+1} = x^k + \alpha^k d^k$.
  - Check for some stopping criterion and break from loop.

Characterization Of Local Optimum

Proposition

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function over the set $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x^\star \in int(\mathcal{C})$ be a local optimum point of f. Let all partial derivatives of f exist at $x^\star$. Then $\nabla f(x^\star) = \mathbf{0}$.

Characterization Of Local Optimum

Proposition

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function over the set $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x^\star \in int(\mathcal{C})$ be a local optimum point of f. Let all partial derivatives of f exist at $x^\star$. Then $\nabla f(x^\star) = \mathbf{0}$.

Proof idea:

Characterization Of Local Optimum

Proposition

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function over the set $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x^\star \in int(\mathcal{C})$ be a local optimum point of f. Let all partial derivatives of f exist at $x^\star$. Then $\nabla f(x^\star) = \mathbf{0}$.

Proof idea:

- Consider $e_i = [0 \ldots 0 \underbrace{1}_{i\text{-th coordinate}} 0 \ldots 0]^\top$ containing 1 at the i-th coordinate.

Characterization Of Local Optimum

Proposition

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function over the set $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x^\star \in int(\mathcal{C})$ be a local optimum point of f. Let all partial derivatives of f exist at $x^\star$. Then $\nabla f(x^\star) = \mathbf{0}$.

Proof idea:

- Consider $e_i = [0 \ldots 0 \underbrace{1}_{i\text{-th coordinate}} 0 \ldots 0]^\top$ containing 1 at the i-th coordinate.
- Let $g(\alpha) = f(x^\star + \alpha e_i)$. Note g is a scalar function.

Characterization Of Local Optimum

Proposition

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function over the set $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x^\star \in int(\mathcal{C})$ be a local optimum point of f. Let all partial derivatives of f exist at $x^\star$. Then $\nabla f(x^\star) = \mathbf{0}$.

Proof idea:

- Consider $e_i = [0 \ldots 0 \underbrace{1}_{i\text{-th coordinate}} 0 \ldots 0]^\top$ containing 1 at the i-th coordinate.
- Let $g(\alpha) = f(x^\star + \alpha e_i)$. Note g is a scalar function.
- $x^\star$ is a local optimum point of $f \implies 0$ is a local optimum point of g. (Why?)

Characterization Of Local Optimum

Proposition

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function over the set $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x^\star \in int(\mathcal{C})$ be a local optimum point of f. Let all partial derivatives of f exist at $x^\star$. Then $\nabla f(x^\star) = \mathbf{0}$.

Proof idea:

- Consider $e_i = [0 \ldots 0 \underbrace{1}_{i\text{-th coordinate}} 0 \ldots 0]^\top$ containing 1 at the i-th coordinate.
- Let $g(\alpha) = f(x^\star + \alpha e_i)$. Note g is a scalar function.
- $x^\star$ is a local optimum point of $f \implies 0$ is a local optimum point of g. (Why?)
- $\implies g'(0) = 0$.

Characterization Of Local Optimum

Proposition

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function over the set $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x^\star \in int(\mathcal{C})$ be a local optimum point of f. Let all partial derivatives of f exist at $x^\star$. Then $\nabla f(x^\star) = \mathbf{0}$.

Proof idea:

- Consider $e_i = [0 \ldots 0 \underbrace{1}_{i\text{-th coordinate}} 0 \ldots 0]^\top$ containing 1 at the i-th coordinate.
- Let $g(\alpha) = f(x^\star + \alpha e_i)$. Note g is a scalar function.
- $x^\star$ is a local optimum point of $f \implies 0$ is a local optimum point of g. (Why?)
- $\implies g'(0) = 0$.
- However $g'(0) = \langle \nabla f(x^\star), e_i \rangle = \frac{\partial f(x^\star)}{\partial x_i} = 0$. (How?)

Characterization Of Local Optimum

Proposition

Let $f : \mathcal{C} \longrightarrow \mathbb{R}$ be a function over the set $\mathcal{C} \subseteq \mathbb{R}^d$. Let $x^\star \in int(\mathcal{C})$ be a local optimum point of f. Let all partial derivatives of f exist at $x^\star$. Then $\nabla f(x^\star) = \mathbf{0}$.

Proof idea:

- Consider $e_i = [0 \ldots 0 \underbrace{1}_{i\text{-th coordinate}} 0 \ldots 0]^\top$ containing 1 at the i-th coordinate.
- Let $g(\alpha) = f(x^\star + \alpha e_i)$. Note g is a scalar function.
- $x^\star$ is a local optimum point of $f \implies 0$ is a local optimum point of g. (Why?)
- $\implies g'(0) = 0$.
- However $g'(0) = \langle \nabla f(x^\star), e_i \rangle = \frac{\partial f(x^\star)}{\partial x_i} = 0$. (How?)
- $\implies \nabla f(x^\star) = \mathbf{0}$. (Why?)

Algorithm Development using Descent Direction

Consider the general optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \qquad \text{(GEN-OPT)}$$

where $f : \mathbb{R}^d \longrightarrow \mathbb{R}$.

Algorithm to solve (GEN-OPT)

- Start with $x^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - Find a descent direction d^k of f at x^k and $\alpha^k > 0$ such that $f(x^k + \alpha^k d^k) < f(x^k)$.
  - $x^{k+1} = x^k + \alpha^k d^k$.
  - If $\|\nabla f(x^{k+1})\|_2 = 0$, set $x^* = x^{k+1}$, break from loop.
- Output x^*.

Algorithm Development using Descent Direction

Algorithm to solve (GEN-OPT)

- Start with $x^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - Find a descent direction d^k of f at x^k and $\alpha^k > 0$ such that $f(x^k + \alpha^k d^k) < f(x^k)$.
  - $x^{k+1} = x^k + \alpha^k d^k$.
  - If $\|\nabla f(x^{k+1})\|_2 = 0$, set $x^* = x^{k+1}$, break from loop.
- Output x^*.

Homework: Compare the structure of this algorithm with the Perceptron training algorithm and try to understand the perceptron update rule from an optimization perspective.

Algorithm Development using Descent Direction

Consider the general optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \qquad \text{(GEN-OPT)}$$

where $f : \mathbb{R}^d \longrightarrow \mathbb{R}$.

Gradient Descent Algorithm to solve (GEN-OPT)

- Start with $x^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - $d^k = -\nabla f(x^k)$.
  - $\alpha^k = \text{argmin}_{\alpha > 0} f(x^k + \alpha d^k)$.
  - $x^{k+1} = x^k + \alpha^k d^k$.
  - If $\|\nabla f(x^{k+1})\|_2 = 0$, set $x^* = x^{k+1}$, break from loop.
- Output x^*.

Gradient Descent for our MLP Problem

Recall: For MLP, the loss minimization problem is:

$$\min_{w=(w_{11}^1, w_{12}^1)} E(w) = \sum_{i=1}^{4} e^i(w) = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

where $E : \mathbb{R}^2 \longrightarrow \mathbb{R}$.

Gradient Descent Algorithm to solve MLP Loss Minimization Problem

- Start with $w^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - $d^k = -\nabla E(w^k)$.
  - $\alpha^k = \text{argmin}_{\alpha > 0} E(w^k + \alpha d^k)$.
  - $w^{k+1} = w^k + \alpha^k d^k$.
  - If $\|\nabla E(w^{k+1})\|_2 = 0$, set $w^* = w^{k+1}$, break from loop.
- Output w^*.

Gradient Descent for our MLP Problem

Recall: For MLP, the loss minimization problem is:

$$\min_{w=(w_{11}^1, w_{12}^1)} E(w) = \sum_{i=1}^{4} e^i(w) = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

Gradient Descent Algorithm to solve MLP Loss Minimization Problem

- Start with $w^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - $d^k = -\nabla E(w^k)$.
  - $\alpha^k = \text{argmin}_{\alpha > 0} E(w^k + \alpha d^k)$.
  - $w^{k+1} = w^k + \alpha^k d^k$.
  - If $\|\nabla E(w^{k+1})\|_2 = 0$, set $w^* = w^{k+1}$, break from loop.
- Output w^*.

Gradient Descent for our MLP Problem

Recall: For MLP, the loss minimization problem is:

$$\min_{w=(w_{11}^1, w_{12}^1)} E(w) = \sum_{i=1}^{4} e^i(w) = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

Gradient Descent Algorithm to solve MLP Loss Minimization Problem

- Start with $w^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - $d^k = -\sum_{i=1}^{4} \nabla e^i(w^k)$.
  - $\alpha^k = \text{argmin}_{\alpha > 0} E(w^k + \alpha d^k)$.
  - $w^{k+1} = w^k + \alpha^k d^k$.
  - If $\|\nabla E(w^{k+1})\|_2 = 0$, set $w^* = w^{k+1}$, break from loop.
- Output w^*.

Gradient Descent for our MLP Problem

Recall: For MLP, the loss minimization problem is:

$$\min_{w=(w_{11}^1, w_{12}^1)} E(w) = \sum_{i=1}^{4} e^i(w) = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

Gradient Descent:

- Function values $E(w^t)$ exhibit $O(1/\sqrt{k})$ convergence under minor assumptions and the assumption of existence of a local optimum.
- $O(1/k^2)$ convergence possible.
- Linear convergence also possible for strongly convex and smooth function $E(w)$.
- Arbitrary accuracy possible $|W(w^{gd}) - E(w^*)| \approx O(10^{-15})$.

Gradient Descent for our MLP Problem

Recall: For MLP, the loss minimization problem is:

$$\min_{w=(w_{11}^1, w_{12}^1)} E(w) = \sum_{i=1}^{4} e^i(w) = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

Gradient Descent:

- Blind to structure of $E(w)$.
- Finding proper α^k at each k is computationally intensive - takes at least $O(Sd)$ time.
- Storage complexity: $O(d)$

Stochastic Gradient Descent for our MLP Problem

Recall: For MLP, the loss minimization problem is:

$$\min_{w=(w_{11}^1, w_{12}^1)} E(w) = \sum_{i=1}^{4} e^i(w) = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

Stochastic Gradient Descent Algorithm to solve MLP Loss Minimization Problem

- Start with $w^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - Choose a sample $j_k \in \{1, \ldots, 4\}$.
  - $w^{k+1} \leftarrow w^k - \gamma_k \nabla_w e^{j_k}(w^k)$.

Regularized Empirical Loss Minimization - Optimization Methods

Stochastic Gradient Descent Algorithm to solve MLP Loss Minimization Problem

- Start with $w^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - Choose a sample $j_k \in \{1, \ldots, 4\}$.
  - $w^{k+1} \leftarrow w^k - \gamma_k \nabla_w e^{j_k}(w^k)$.

$\nabla_w e^{j_k}(w^k)$: Gradient at point w^k, of e^{i_k} with respect to w. Takes only $O(d)$ time.

Under suitable conditions on γ_k ($\sum_k \gamma_k^2 < \infty$, $\sum_k \gamma_k \to \infty$), this procedure converges **asymptotically**.
For smooth functions, $O(1/k)$ convergence possible (in theory!).
Typical choice: $\gamma_k = \frac{1}{k+1}$.

Mini-Batch Stochastic Gradient Descent for our MLP Problem

Mini-batch SGD Algorithm to solve MLP Loss Minimization Problem

- Start with $w^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - Choose a block of samples $B_k \subseteq \{1, \ldots, 4\}$.
  - $w^{k+1} \leftarrow w^k - \gamma_k \sum_{j \in B_k} \nabla_w e^j(w^k)$.

Mini-batch Stochastic Gradient Descent for our MLP Problem

Mini-batch SGD Algorithm to solve MLP Loss Minimization Problem

- Start with $w^0 \in \mathbb{R}^d$.
- For $k = 0, 1, 2, \ldots$
  - Choose a block of samples $B_k \subseteq \{1, \ldots, 4\}$.
  - $w^{k+1} \leftarrow w^k - \gamma_k \sum_{j \in B_k} \nabla_w e^j(w^k)$.

- Restrictions on γ_k similar to that in SGD.
- **Asymptotic convergence** !

GD/SGD: Crucial Step

Recall: For MLP, the loss minimization problem is:

$$\min_{w=(w_{11}^1, w_{12}^1)} E(w) = \sum_{i=1}^{4} e^i(w) = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

Crucial step in Gradient Descent Algorithm

$w^{k+1} = w^k - \alpha^k \sum_{i=1}^{4} \nabla e^i(w^k)$

Crucial step in Stochastic Gradient Descent Algorithm

$w^{k+1} \leftarrow w^k - \gamma_k \nabla_w e^{j_k}(w^k)$.

Crucial step in Mini-batch SGD Algorithm

$w^{k+1} \leftarrow w^k - \gamma_k \sum_{j \in B_k} \nabla_w e^j(w^k)$.

GD/SGD for MLP: Crucial Step

Recall: For MLP, the loss minimization problem is:

$$\min_{w=(w_{11}^1, w_{12}^1)} E(w) = \sum_{i=1}^{4} e^i(w) = \sum_{i=1}^{4} \left(y^i - \frac{1}{1 + \exp\left(-\left[w_{11}^1 x_1^i + w_{12}^1 x_2^i\right]\right)} \right)^2$$

Crucial step in Gradient Descent Algorithm

$w^{k+1} = w^k - \alpha^k \sum_{i=1}^{4} \nabla e^i(w^k)$

Crucial step in Stochastic Gradient Descent Algorithm

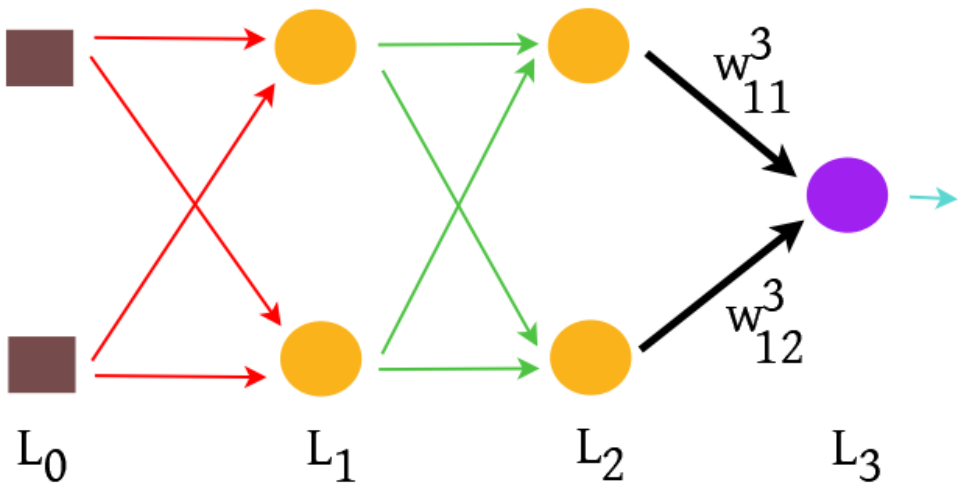
$w^{k+1} \leftarrow w^k - \gamma_k \nabla_w e^{j_k}(w^k)$.

Crucial step in Mini-batch SGD Algorithm

$w^{k+1} \leftarrow w^k - \gamma_k \sum_{j \in B_k} \nabla_w e^j(w^k)$.

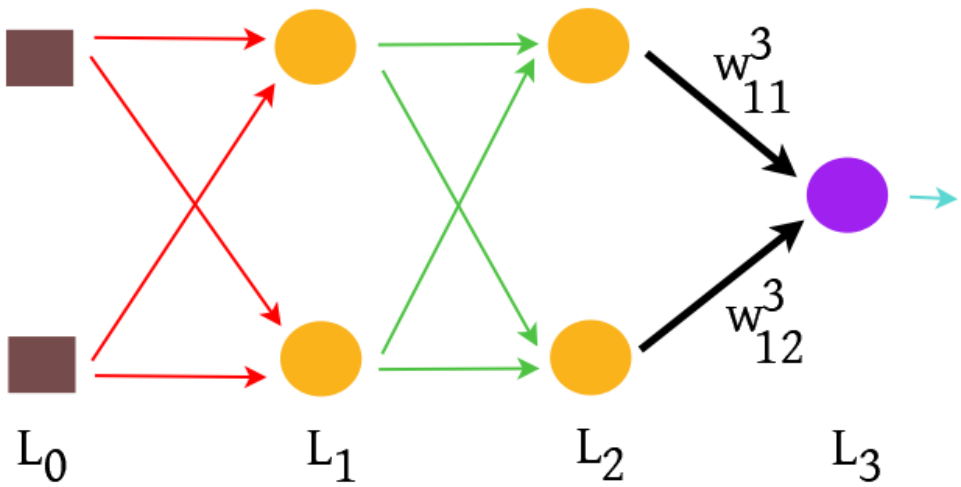
Note: $\nabla e^i(w^k)$, $\nabla_w e^{j_k}(w^k)$, $\nabla e^j(w^k)$ denote sample-wise gradient computation.

GD/SGD for MLP: Sample-wise Gradient Computation



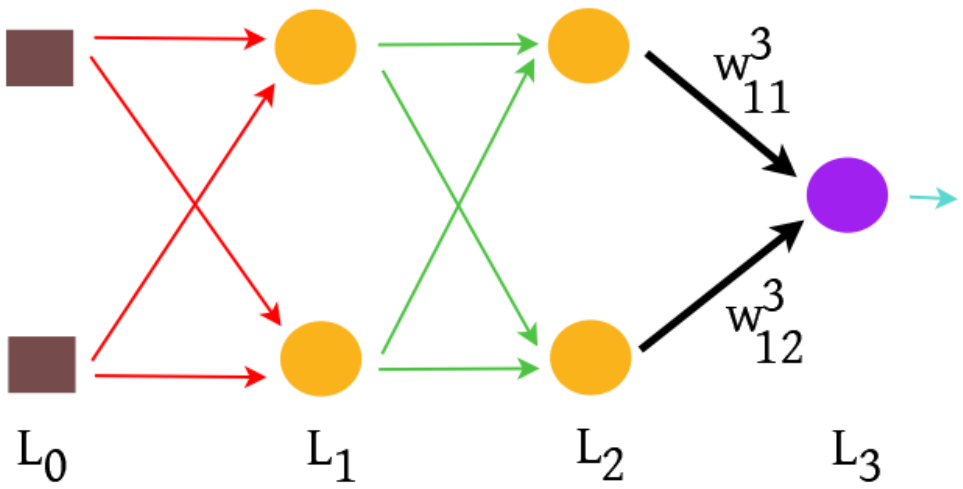
- Consider an arbitrary training sample $(x, y) \in D$.

GD/SGD for MLP: Sample-wise Gradient Computation



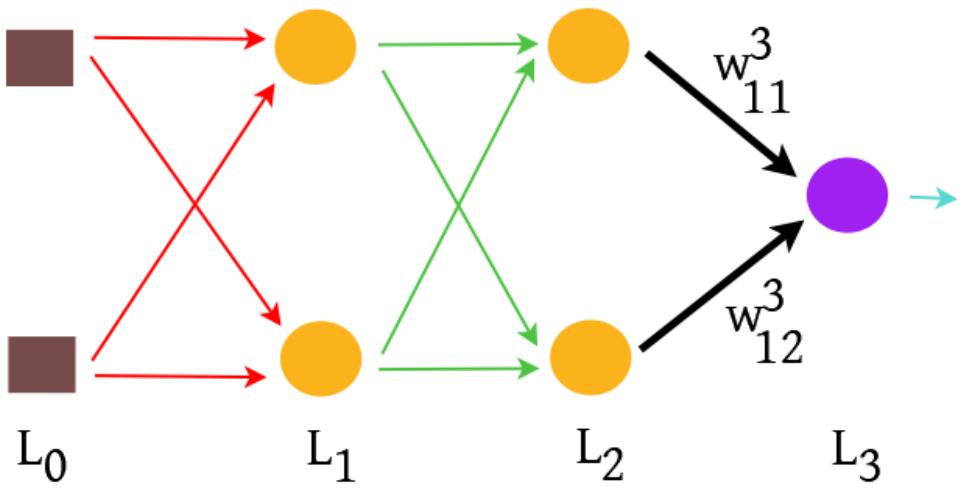
- Consider an arbitrary training sample $(x, y) \in D$.
- At layer L_3, $\hat{y} = a_1^3 = \phi\left(z_1^3\right) = \phi\left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.

GD/SGD for MLP: Sample-wise Gradient Computation



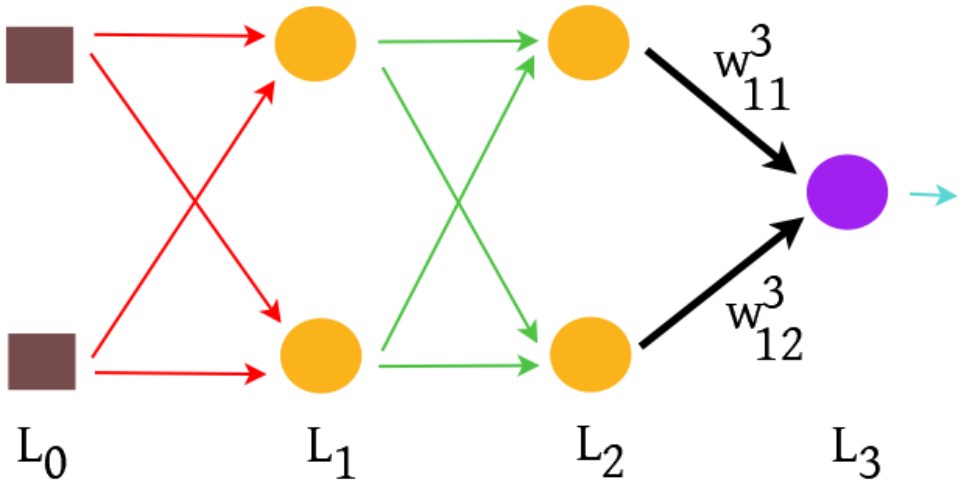
- Consider an arbitrary training sample $(x, y) \in D$.
- At layer L_3, $\hat{y} = a_1^3 = \phi\left(z_1^3\right) = \phi\left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Sample-wise error: $e = (\hat{y} - y)^2$.

GD/SGD for MLP: Sample-wise Gradient Computation



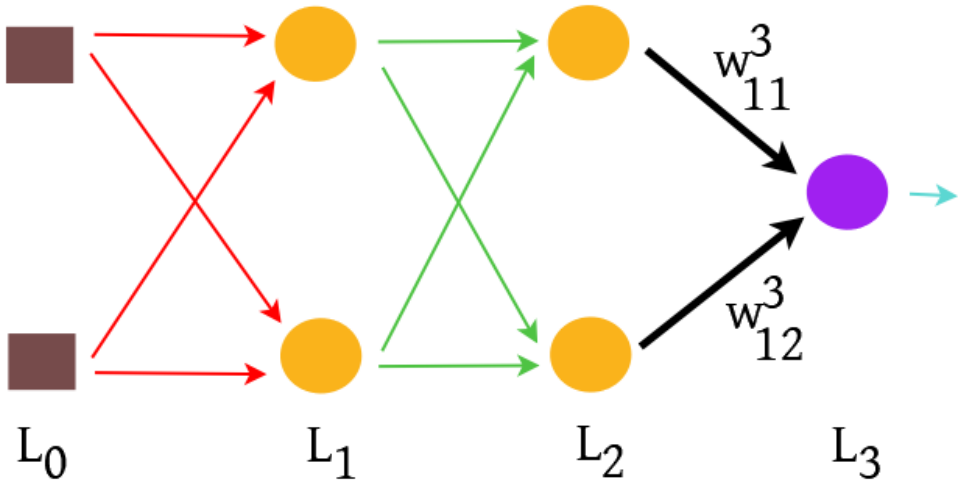
- Consider an arbitrary training sample $(x, y) \in D$.
- At layer L_3, $\hat{y} = a_1^3 = \phi\left(z_1^3\right) = \phi\left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Sample-wise error: $e = (\hat{y} - y)^2$.
- **Aim:** To find $\nabla_w e = [\nabla_{w_{11}^1} e \ \nabla_{w_{12}^1} e \ \ldots \ \nabla_{w_{12}^3} e]^\top$.

GD/SGD for MLP: Sample-wise Gradient Computation



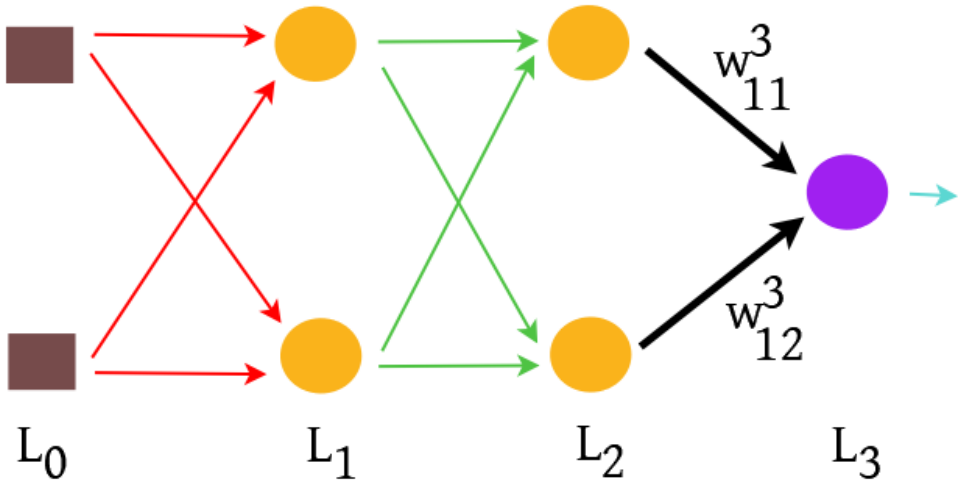
- Consider an arbitrary training sample $(x, y) \in D$.
- At layer L_3, $\hat{y} = a_1^3 = \phi\left(z_1^3\right) = \phi\left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Sample-wise error: $e = (\hat{y} - y)^2$.
- **Note:** $\nabla_{w_{11}^3} e = \frac{\partial e}{\partial z_1^3} \frac{\partial z_1^3}{\partial w_{11}^3}$.

GD/SGD for MLP: Sample-wise Gradient Computation



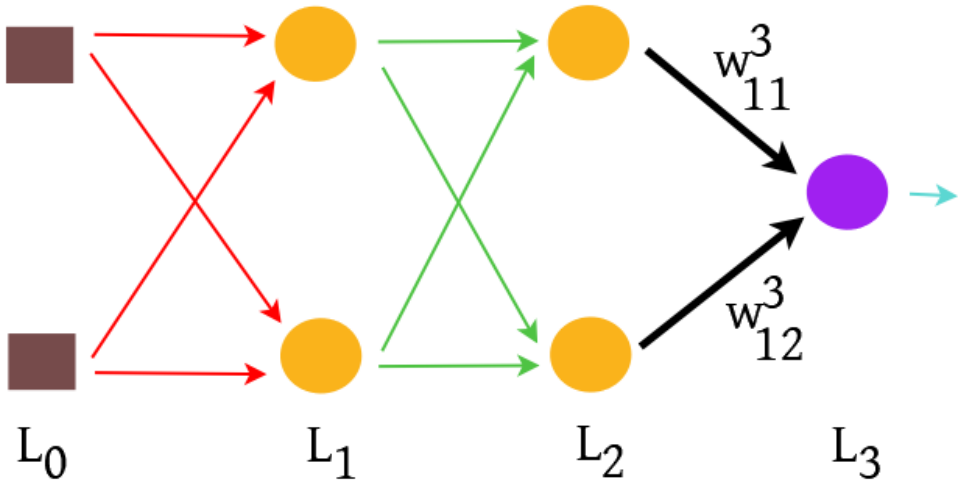
- Consider an arbitrary training sample $(x, y) \in D$.
- At layer L_3, $\hat{y} = a_1^3 = \phi\left(z_1^3\right) = \phi\left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Sample-wise error: $e = (\hat{y} - y)^2$.
- **Note:** $\nabla_{w_{11}^3} e = \frac{\partial e}{\partial z_1^3} \frac{\partial z_1^3}{\partial w_{11}^3} = \frac{\partial e}{\partial z_1^3} a_1^2$.

GD/SGD for MLP: Sample-wise Gradient Computation



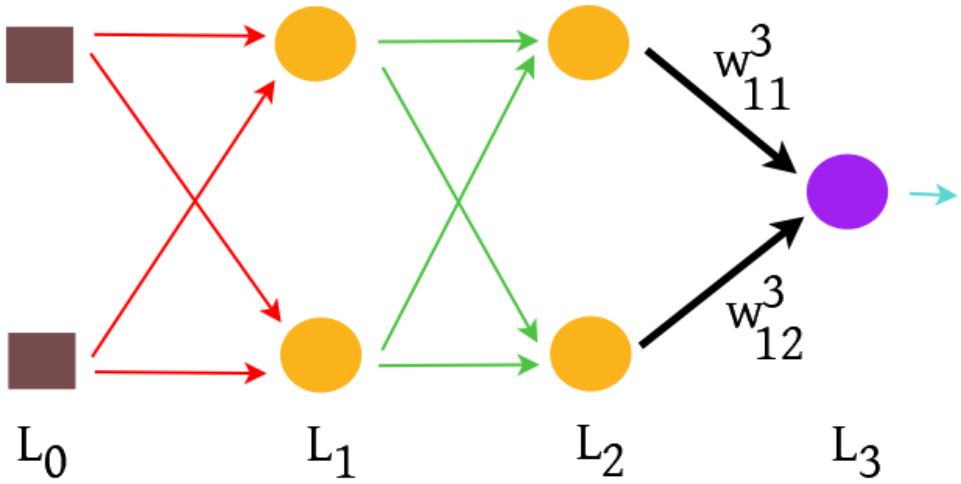
- Consider an arbitrary training sample $(x, y) \in D$.
- At layer L_3, $\hat{y} = a_1^3 = \phi\left(z_1^3\right) = \phi\left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Sample-wise error: $e = (\hat{y} - y)^2$.
- **Note:** $\nabla_{w_{11}^3} e = \frac{\partial e}{\partial z_1^3} \frac{\partial z_1^3}{\partial w_{11}^3} = \frac{\partial e}{\partial a_1^3} \frac{\partial a_1^3}{\partial z_1^3} a_1^2$.

GD/SGD for MLP: Sample-wise Gradient Computation



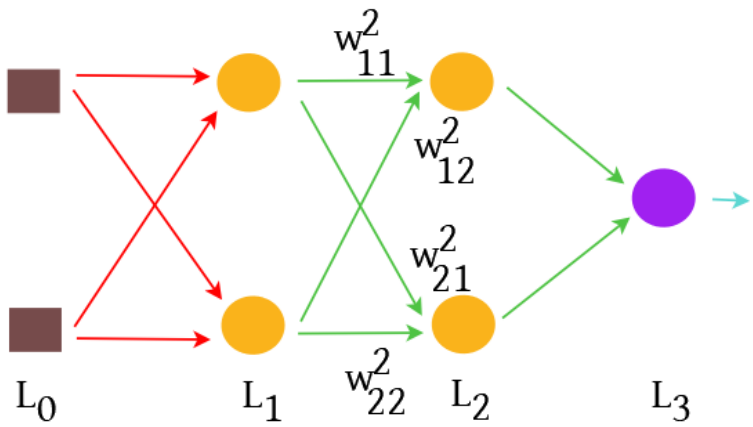
- Consider an arbitrary training sample $(x, y) \in D$.
- At layer L_3, $\hat{y} = a_1^3 = \phi\left(z_1^3\right) = \phi\left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Sample-wise error: $e = (\hat{y} - y)^2$.
- **Note:** $\nabla_{w_{11}^3} e = \frac{\partial e}{\partial z_1^3}\frac{\partial z_1^3}{\partial w_{11}^3} = \frac{\partial e}{\partial a_1^3}\frac{\partial a_1^3}{\partial z_1^3}a_1^2 = \frac{\partial e}{\partial \hat{y}}\phi'(z_1^3)a_1^2$.

GD/SGD for MLP: Sample-wise Gradient Computation



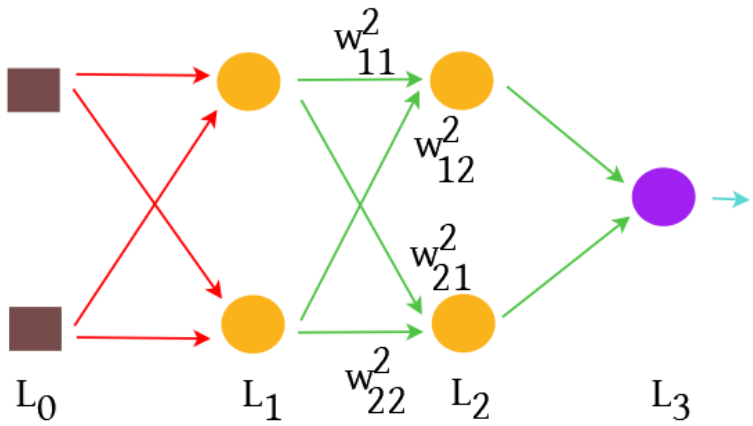
- Consider an arbitrary training sample $(x, y) \in D$.
- At layer L_3, $\hat{y} = a_1^3 = \phi\left(z_1^3\right) = \phi\left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Sample-wise error: $e = (\hat{y} - y)^2$.
- **Note:** $\nabla_{w_{11}^3} e = \frac{\partial e}{\partial z_1^3}\frac{\partial z_1^3}{\partial w_{11}^3} = \frac{\partial e}{\partial a_1^3}\frac{\partial a_1^3}{\partial z_1^3}a_1^2 = \frac{\partial e}{\partial \hat{y}}\phi'(z_1^3)a_1^2$.
- Similarly, $\nabla_{w_{12}^3} e = \frac{\partial e}{\partial \hat{y}}\phi'(z_1^3)a_2^2$.

GD/SGD for MLP: Sample-wise Gradient Computation



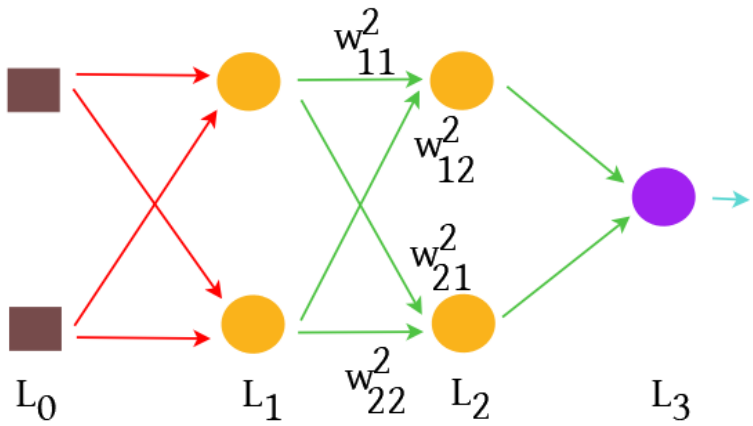
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.

GD/SGD for MLP: Sample-wise Gradient Computation



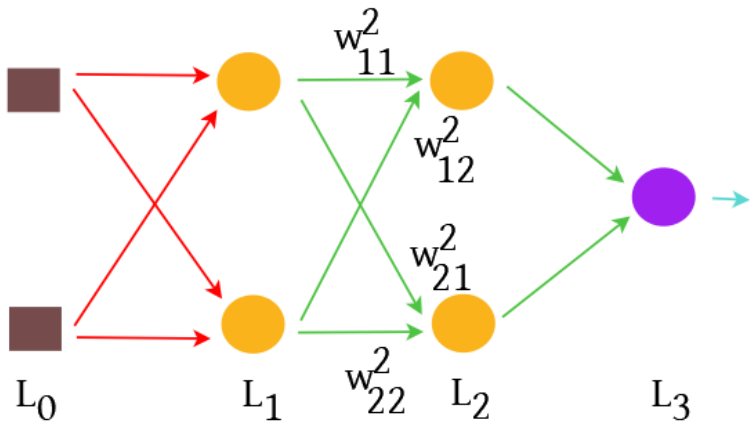
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial e}{\partial z_1^2} a_1^1$.

GD/SGD for MLP: Sample-wise Gradient Computation



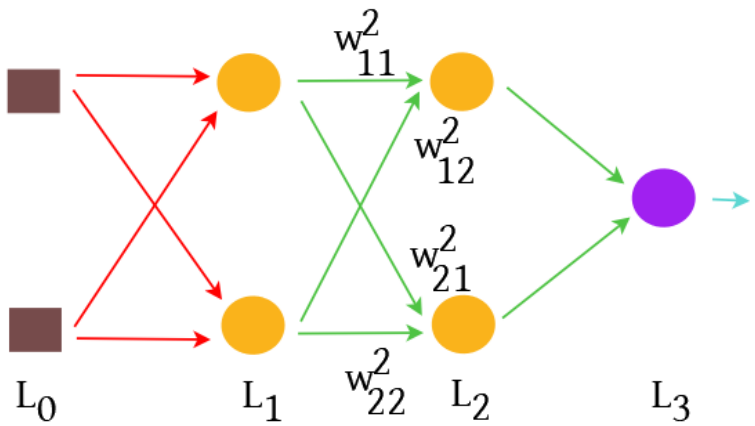
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial e}{\partial z_1^2} a_1^1$.

GD/SGD for MLP: Sample-wise Gradient Computation



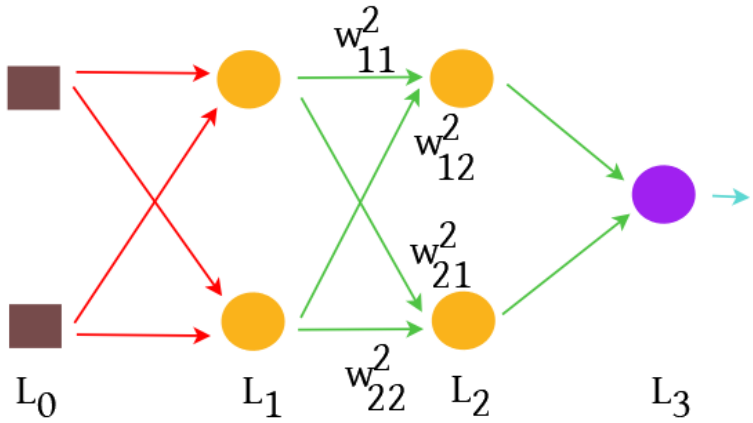
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial e}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\frac{\partial a_1^2}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\phi'(z_1^2) a_1^1$.

GD/SGD for MLP: Sample-wise Gradient Computation



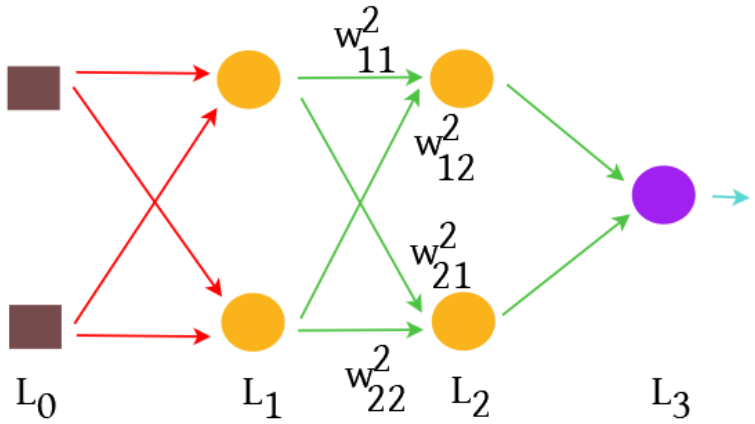
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial e}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\frac{\partial a_1^2}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\phi'(z_1^2) a_1^1$.
- Now recall that $z_1^3 = \left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.

GD/SGD for MLP: Sample-wise Gradient Computation



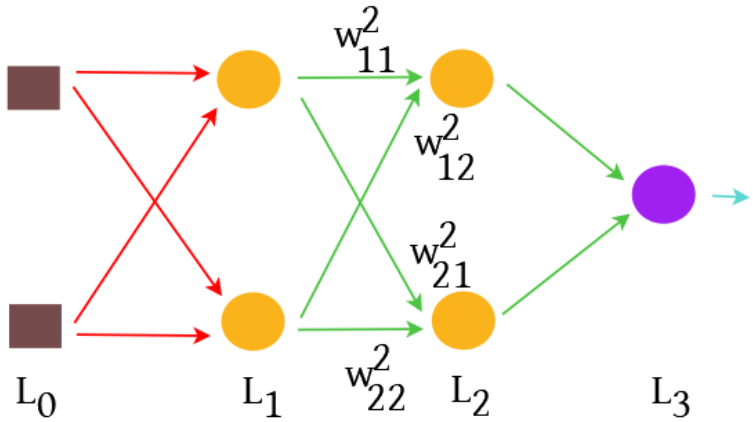
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial e}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\frac{\partial a_1^2}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\phi'(z_1^2) a_1^1$.
- Now recall that $z_1^3 = \left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Hence $\frac{\partial e}{\partial a_1^2} = \frac{\partial e}{\partial z_1^3}\frac{\partial z_1^3}{\partial a_1^2} = \frac{\partial e}{\partial z_1^3} w_{11}^3$.

GD/SGD for MLP: Sample-wise Gradient Computation



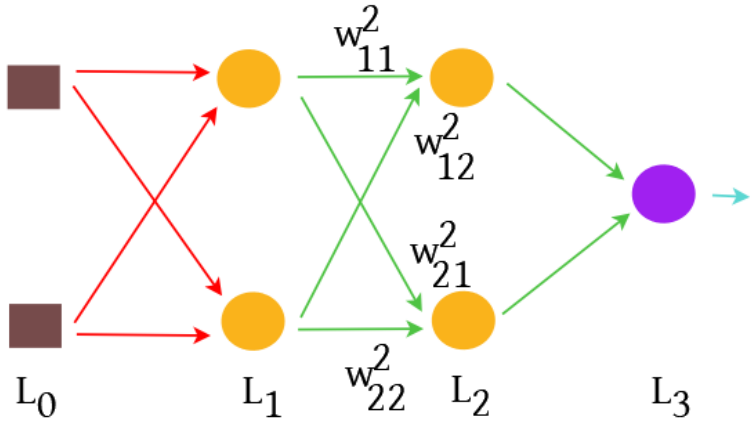
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial e}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\frac{\partial a_1^2}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\phi'(z_1^2) a_1^1$.
- Now recall that $z_1^3 = \left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Hence $\frac{\partial e}{\partial a_1^2} = \frac{\partial e}{\partial z_1^3}\frac{\partial z_1^3}{\partial a_1^2} = \frac{\partial e}{\partial z_1^3} w_{11}^3$.
- Recall: We have already computed $\frac{\partial e}{\partial z_1^3} = \frac{\partial e}{\partial \hat{y}}\phi'(z_1^3)$.

GD/SGD for MLP: Sample-wise Gradient Computation



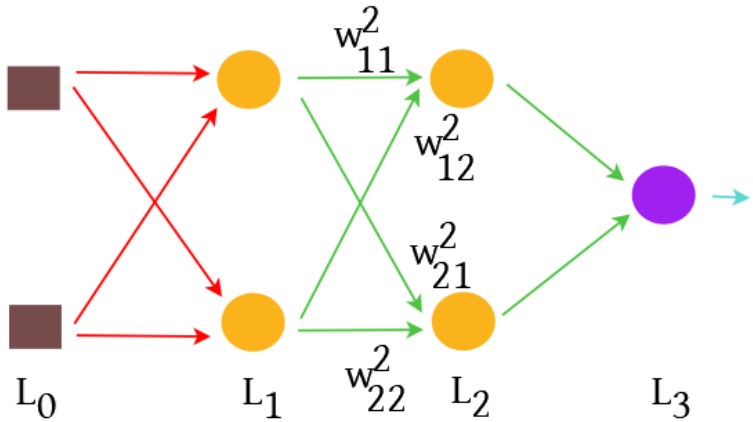
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial e}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\frac{\partial a_1^2}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\phi'(z_1^2) a_1^1$.
- Now recall that $z_1^3 = \left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Hence $\frac{\partial e}{\partial a_1^2} = \frac{\partial e}{\partial z_1^3}\frac{\partial z_1^3}{\partial a_1^2} = \frac{\partial e}{\partial z_1^3} w_{11}^3 = \frac{\partial e}{\partial \hat{y}}\phi'(z_1^3) w_{11}^3$.

GD/SGD for MLP: Sample-wise Gradient Computation



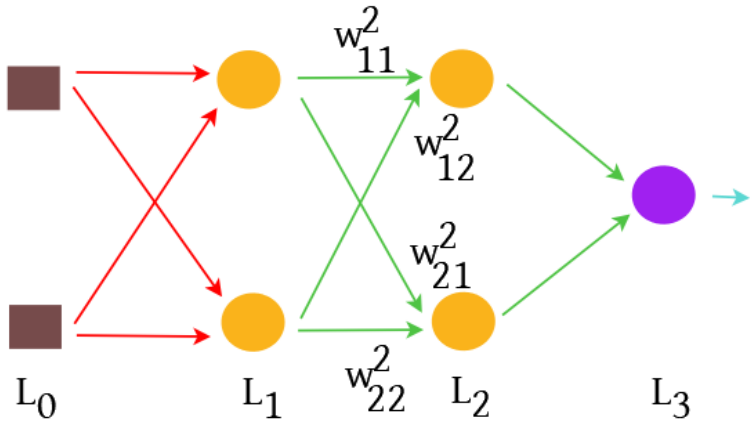
- We have at layer L_2: $a_1^2 = \phi\left(z_1^2\right) = \phi\left(w_{11}^2 a_1^1 + w_{12}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial e}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\frac{\partial a_1^2}{\partial z_1^2} a_1^1 = \frac{\partial e}{\partial a_1^2}\phi'(z_1^2) a_1^1$.
- Now recall that $z_1^3 = \left(w_{11}^3 a_1^2 + w_{12}^3 a_2^2\right)$.
- Hence $\frac{\partial e}{\partial a_1^2} = \frac{\partial e}{\partial z_1^3}\frac{\partial z_1^3}{\partial a_1^2} = \frac{\partial e}{\partial z_1^3} w_{11}^3 = \frac{\partial e}{\partial \hat{y}}\phi'(z_1^3) w_{11}^3$.
- Combining, we have $\nabla_{w_{11}^2} e = \frac{\partial e}{\partial \hat{y}}\phi'(z_1^3) w_{11}^3 \phi'(z_1^2) a_1^1$.

GD/SGD for MLP: Sample-wise Gradient Computation



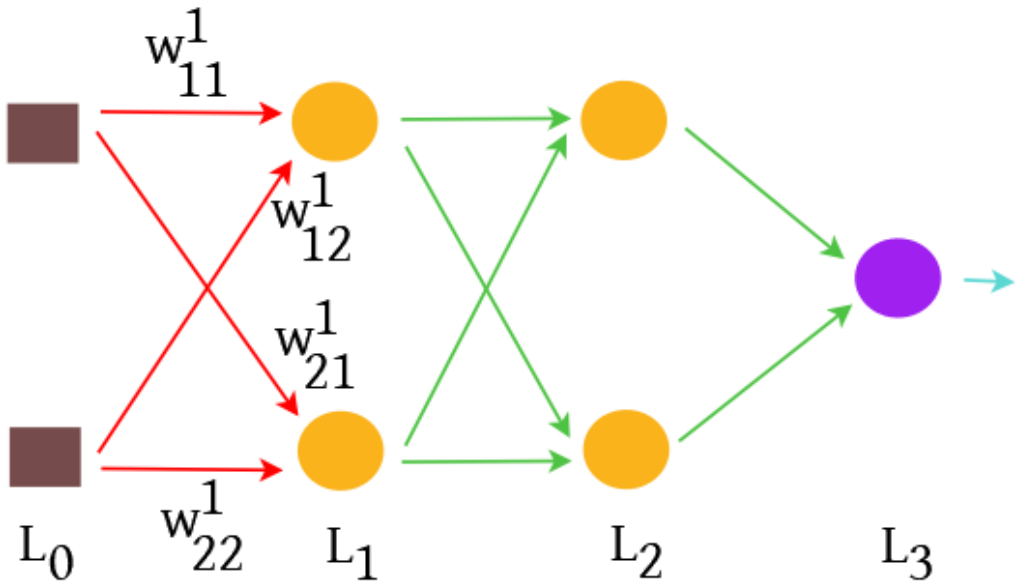
- Thus, $\nabla_{w^2_{11}} e = \frac{\partial e}{\partial \hat{y}} \phi'(z^3_1) w^3_{11} \phi'(z^2_1) a^1_1$.
- Similarly, $\nabla_{w^2_{12}} e = \frac{\partial e}{\partial \hat{y}} \phi'(z^3_1) w^3_{11} \phi'(z^2_1) a^1_2$.

GD/SGD for MLP: Sample-wise Gradient Computation



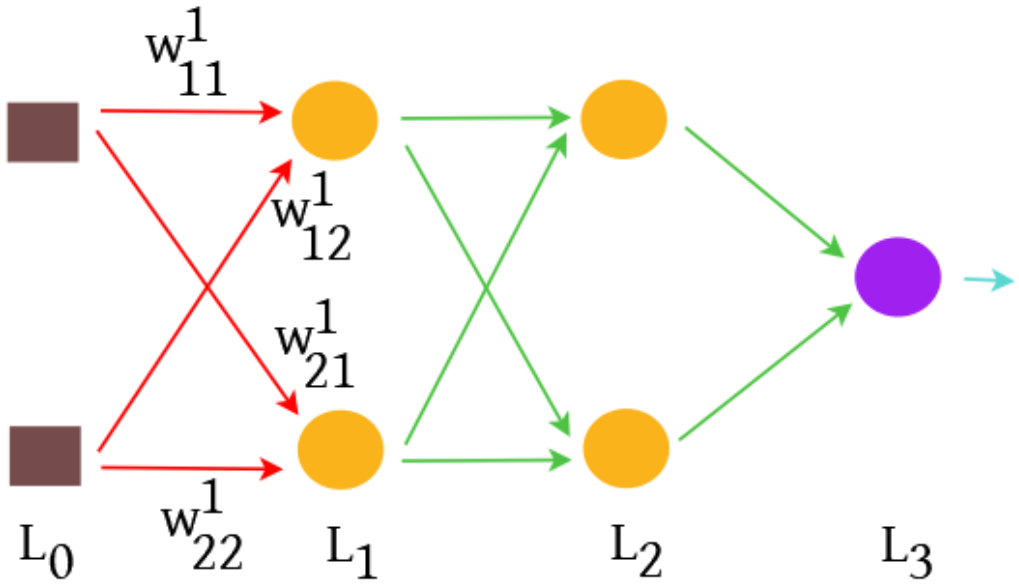
- Also, we have at layer L_2: $a_2^2 = \phi\left(z_2^2\right) = \phi\left(w_{21}^2 a_1^1 + w_{22}^2 a_2^1\right)$.
- Hence, $\nabla_{w_{21}^2} e = ?$, $\nabla_{w_{22}^2} e = ?$

GD/SGD for MLP: Sample-wise Gradient Computation



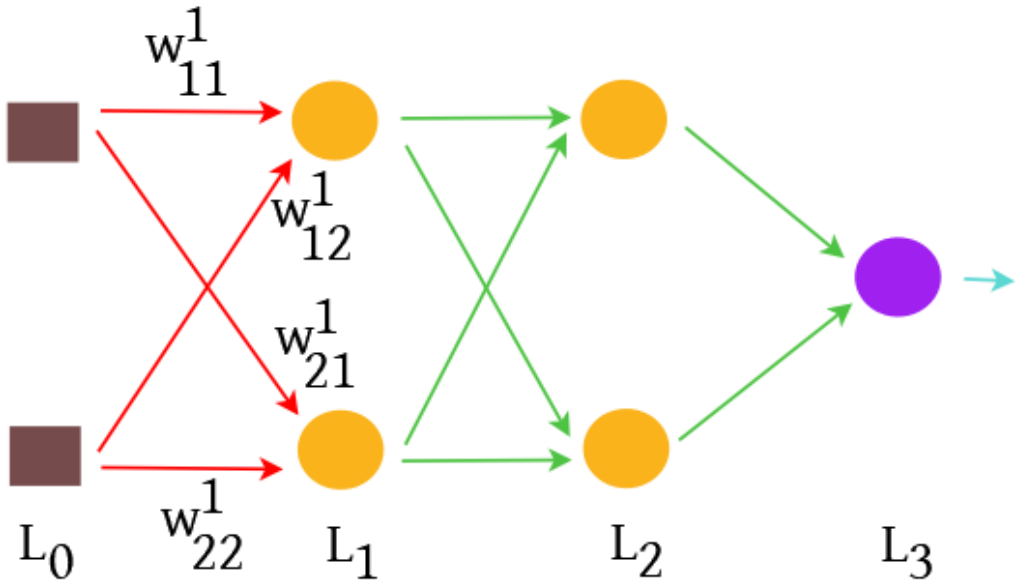
- We have at layer L_1: $a_1^1 = \phi\left(z_1^1\right) = \phi\left(w_{11}^1 x_1 + w_{12}^1 x_2\right)$.

GD/SGD for MLP: Sample-wise Gradient Computation



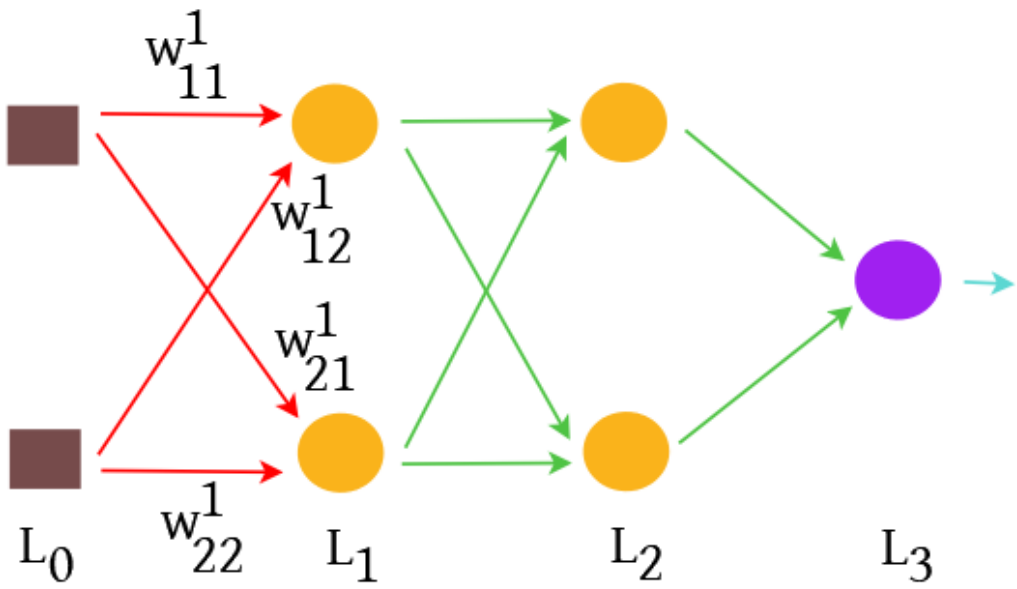
- We have at layer L_1: $a_1^1 = \phi\left(z_1^1\right) = \phi\left(w_{11}^1 x_1 + w_{12}^1 x_2\right)$.
- **Note:** $\nabla_{w_{11}^1} e = \frac{\partial e}{\partial z_1^1}\frac{\partial z_1^1}{\partial w_{11}^1} = \frac{\partial e}{\partial z_1^1} x_1$.

GD/SGD for MLP: Sample-wise Gradient Computation



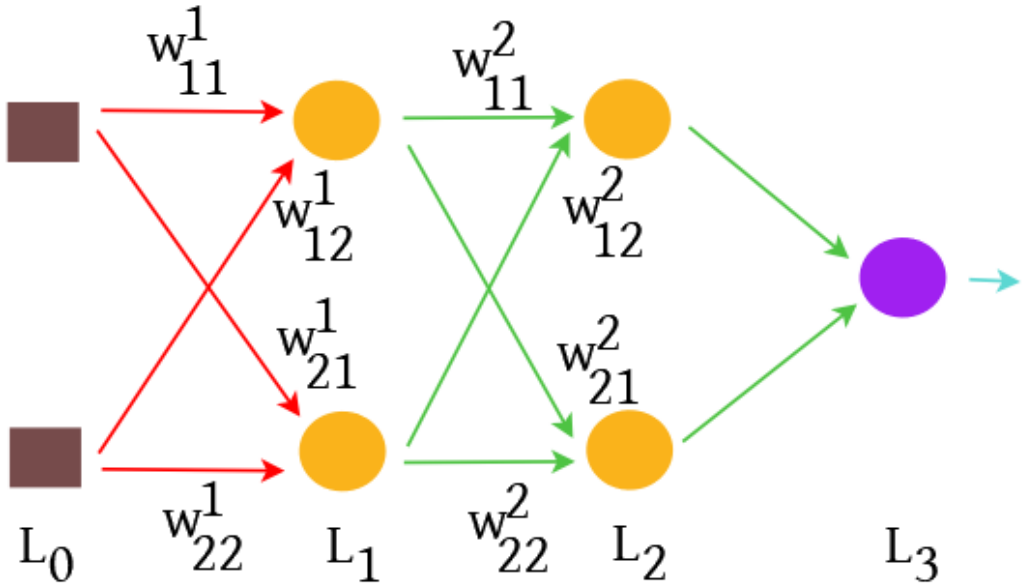
- We have at layer L_1: $a_1^1 = \phi\left(z_1^1\right) = \phi\left(w_{11}^1 x_1 + w_{12}^1 x_2\right)$.
- **Note:** $\nabla_{w_{11}^1} e = \frac{\partial e}{\partial z_1^1} x_1 = \frac{\partial e}{\partial a_1^1} \phi'(z_1^1) x_1$.

GD/SGD for MLP: Sample-wise Gradient Computation



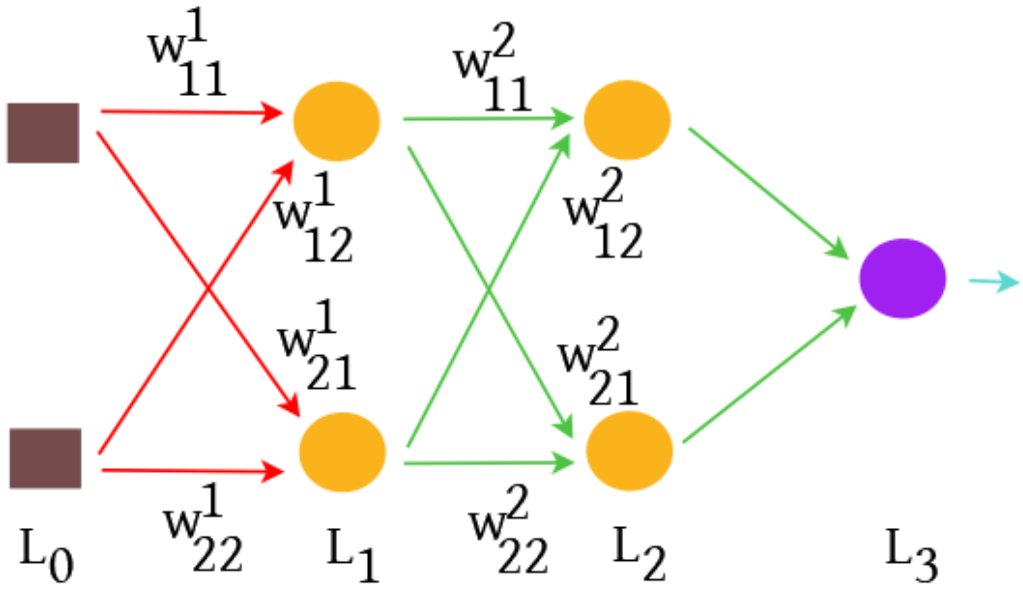
- We have at layer L_1: $a_1^1 = \phi\left(z_1^1\right) = \phi\left(w_{11}^1 x_1 + w_{12}^1 x_2\right)$.
- **Note:** $\nabla_{w_{11}^1} e = \frac{\partial e}{\partial z_1^1} x_1 = \frac{\partial e}{\partial a_1^1} \phi'(z_1^1) x_1$.
- Now we see that a_1^1 contributes to both z_1^2 and z_2^2.

GD/SGD for MLP: Sample-wise Gradient Computation



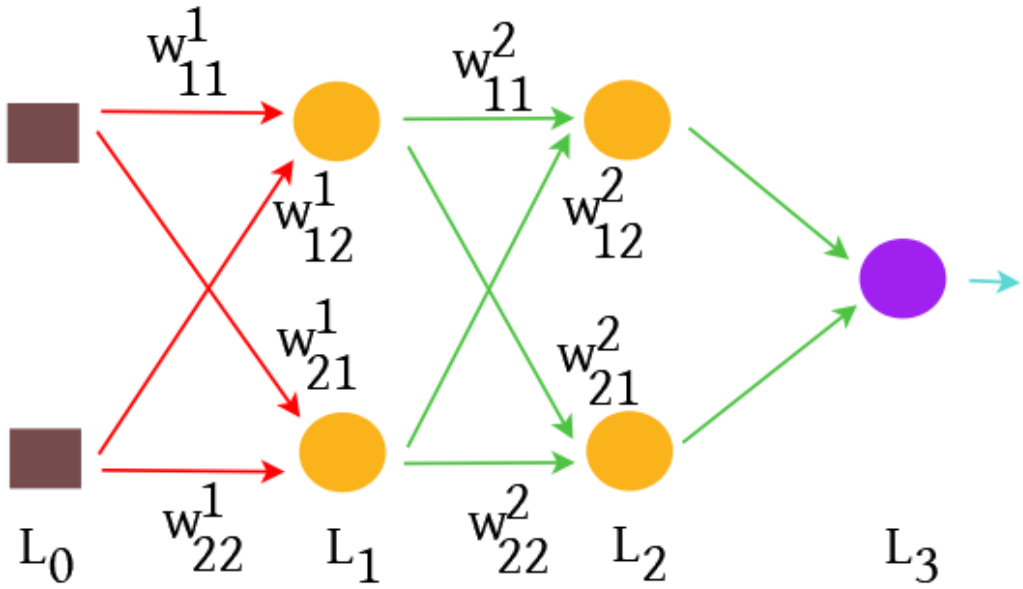
- We have at layer L_1: $a_1^1 = \phi\left(z_1^1\right) = \phi\left(w_{11}^1 x_1 + w_{12}^1 x_2\right)$.
- **Note:** $\nabla_{w_{11}^1} e = \frac{\partial e}{\partial z_1^1} x_1 = \frac{\partial e}{\partial a_1^1} \phi'(z_1^1) x_1$.
- Now we see that a_1^1 contributes to both z_1^2 and z_2^2.
- **Recall:** $z_1^2 = w_{11}^2 a_1^1 + w_{12}^2 a_2^1$ and $z_2^2 = w_{21}^2 a_1^1 + w_{22}^2 a_2^1$.

GD/SGD for MLP: Sample-wise Gradient Computation



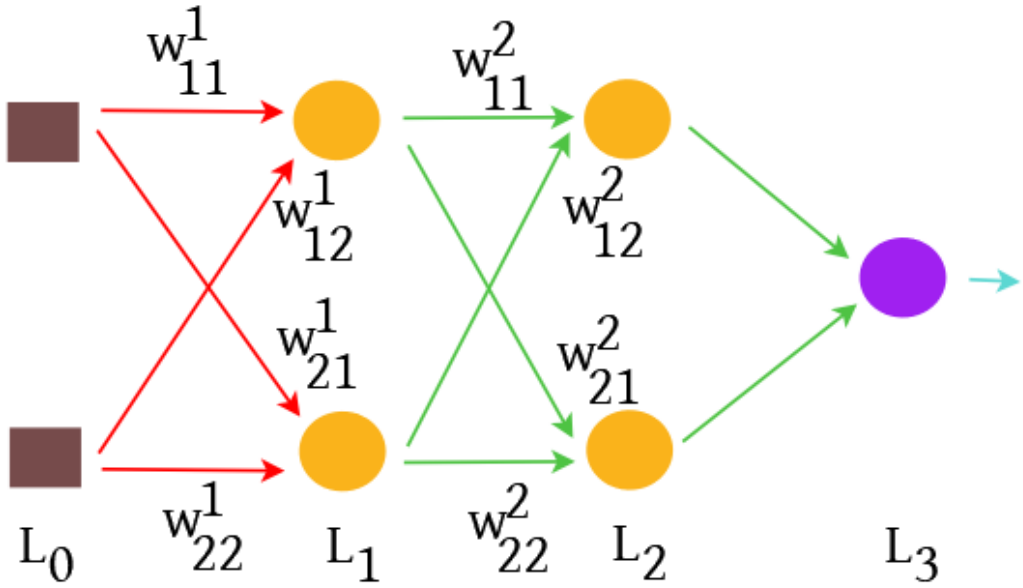
- We have at layer L_1: $a^1_1 = \phi\left(z^1_1\right) = \phi\left(w^1_{11}x_1 + w^1_{12}x_2\right)$.
- **Note:** $\nabla_{w^1_{11}} e = \frac{\partial e}{\partial z^1_1} x_1 = \frac{\partial e}{\partial a^1_1}\phi'(z^1_1)x_1$.
- Now we see that a^1_1 contributes to both z^2_1 and z^2_2.
- **Recall:** $z^2_1 = w^2_{11}a^1_1 + w^2_{12}a^1_2$ and $z^2_2 = w^2_{21}a^1_1 + w^2_{22}a^1_2$.
- Hence $\frac{\partial e}{\partial a^1_1} = \sum_{i=1}^{2} \frac{\partial e}{\partial z^2_i}\frac{\partial z^2_i}{\partial a^1_1}$.

GD/SGD for MLP: Sample-wise Gradient Computation



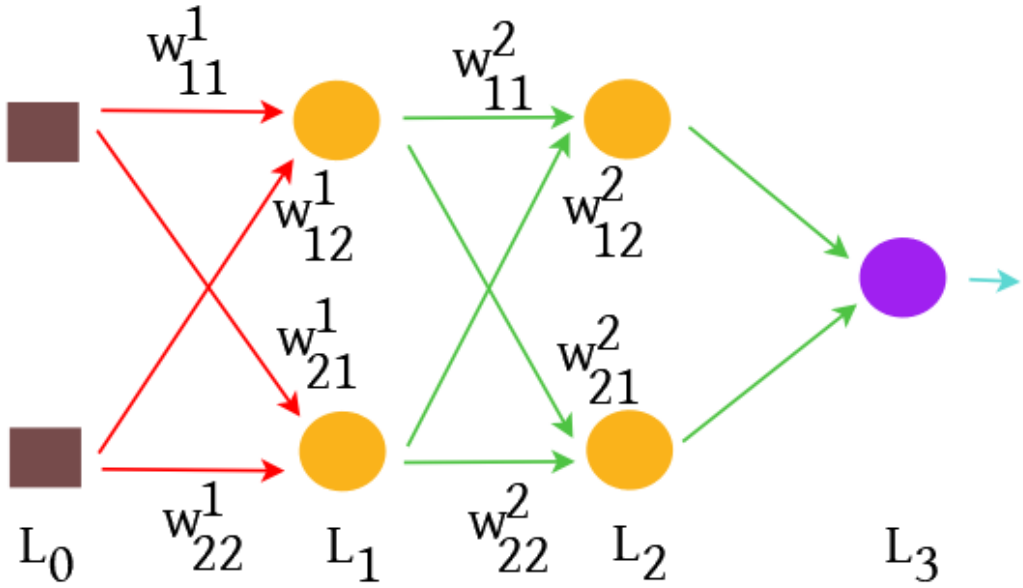
- We have at layer L_1: $a_1^1 = \phi\left(z_1^1\right) = \phi\left(w_{11}^1 x_1 + w_{12}^1 x_2\right)$.
- **Note:** $\nabla_{w_{11}^1} e = \frac{\partial e}{\partial z_1^1} x_1 = \frac{\partial e}{\partial a_1^1} \phi'(z_1^1) x_1$.
- Now we see that a_1^1 contributes to both z_1^2 and z_2^2.
- **Recall:** $z_1^2 = w_{11}^2 a_1^1 + w_{12}^2 a_2^1$ and $z_2^2 = w_{21}^2 a_1^1 + w_{22}^2 a_2^1$.
- Hence $\frac{\partial e}{\partial a_1^1} = \sum_{i=1}^2 \frac{\partial e}{\partial z_i^2} \frac{\partial z_i^2}{\partial a_1^1} = \sum_{i=1}^2 \frac{\partial e}{\partial z_i^2} w_{i1}^2$.

GD/SGD for MLP: Sample-wise Gradient Computation



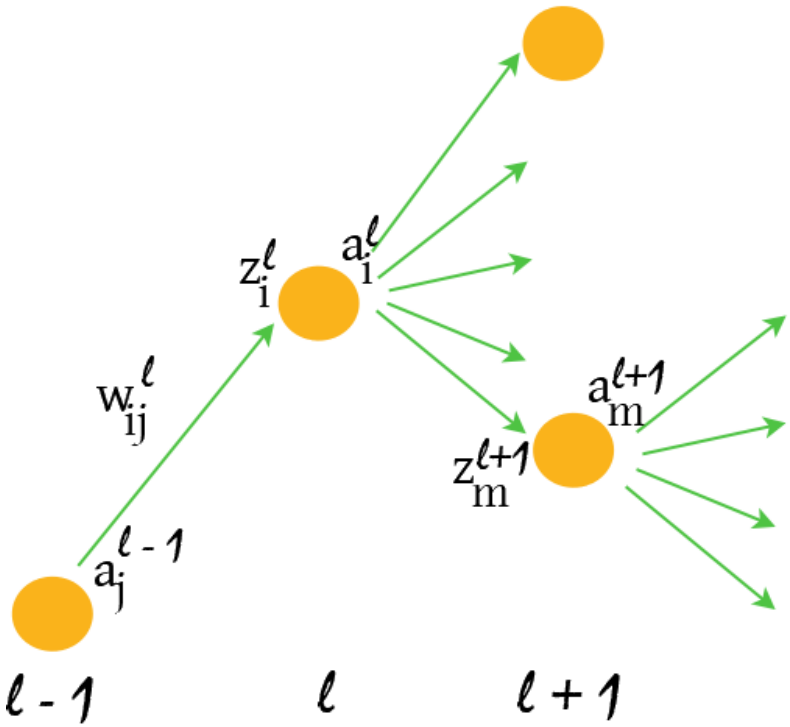
- We have at layer L_1: $a^1_1 = \phi\left(z^1_1\right) = \phi\left(w^1_{11}x_1 + w^1_{12}x_2\right)$.
- **Note:** $\nabla_{w^1_{11}} e = \frac{\partial e}{\partial z^1_1}x_1 = \frac{\partial e}{\partial a^1_1}\phi'(z^1_1)x_1$.
- Now we see that a^1_1 contributes to both z^2_1 and z^2_2.
- **Recall:** $z^2_1 = w^2_{11}a^1_1 + w^2_{12}a^1_2$ and $z^2_2 = w^2_{21}a^1_1 + w^2_{22}a^1_2$.
- Hence $\frac{\partial e}{\partial a^1_1} = \sum_{i=1}^{2} \frac{\partial e}{\partial z^2_i}\frac{\partial z^2_i}{\partial a^1_1} = \sum_{i=1}^{2} \frac{\partial e}{\partial z^2_i}w^2_{i1}$.
- Recall: We have already computed $\frac{\partial e}{\partial z^2_i}, i = 1, 2$.

GD/SGD for MLP: Sample-wise Gradient Computation



- We have at layer L_1: $a^1_1 = \phi\left(z^1_1\right) = \phi\left(w^1_{11}x_1 + w^1_{12}x_2\right)$.
- **Note:** $\nabla_{w^1_{11}} e = \frac{\partial e}{\partial z^1_1} x_1 = \frac{\partial e}{\partial a^1_1}\phi'(z^1_1)x_1$.
- Now we see that a^1_1 contributes to both z^2_1 and z^2_2.
- **Recall:** $z^2_1 = w^2_{11}a^1_1 + w^2_{12}a^1_2$ and $z^2_2 = w^2_{21}a^1_1 + w^2_{22}a^1_2$.
- Hence $\frac{\partial e}{\partial a^1_1} = \sum_{i=1}^{2} \frac{\partial e}{\partial z^2_i}\frac{\partial z^2_i}{\partial a^1_1} = \sum_{i=1}^{2} \frac{\partial e}{\partial z^2_i} w^2_{i1}$.
- Recall: We have already computed $\frac{\partial e}{\partial z^2_i} = \frac{\partial e}{\partial a^2_i}\phi'(z^2_i), i = 1, 2$.

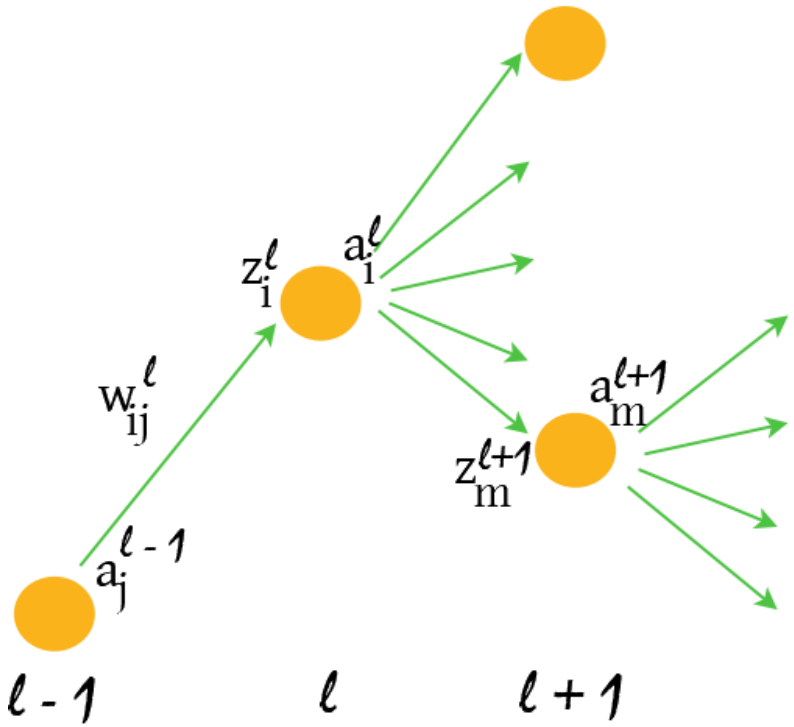
GD/SGD for MLP: Sample-wise Gradient Computation



Generalized setting:

$$\frac{\partial e}{\partial w_{ij}^\ell} = \frac{\partial e}{\partial z_i^\ell} a_j^{\ell-1}$$

GD/SGD for MLP: Sample-wise Gradient Computation

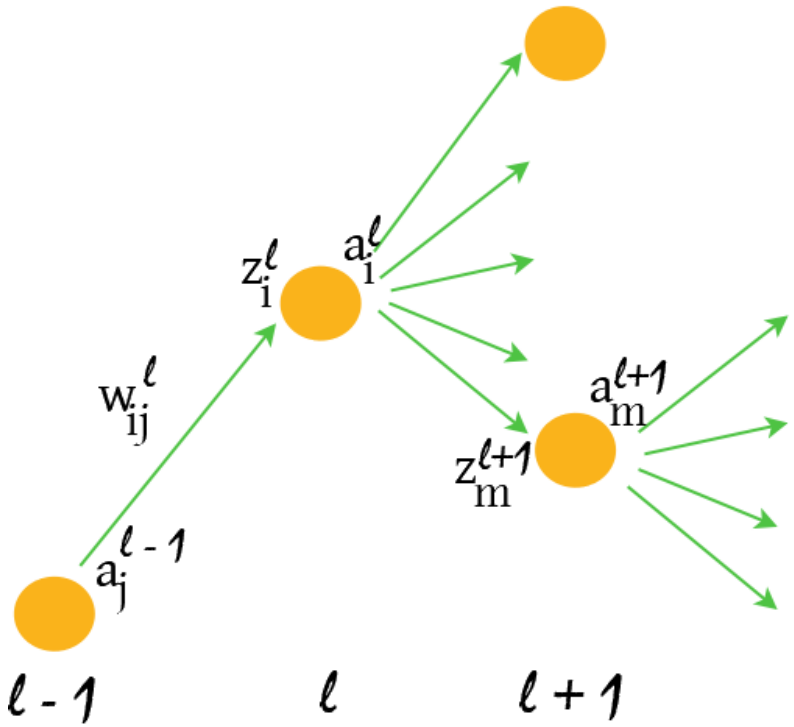


Generalized setting:

$$\frac{\partial e}{\partial w_{ij}^\ell} = \frac{\partial e}{\partial z_i^\ell} a_j^{\ell-1}$$

$$\frac{\partial e}{\partial z_i^\ell} = \frac{\partial e}{\partial a_i^\ell} \phi'(z_i^\ell)$$

GD/SGD for MLP: Sample-wise Gradient Computation



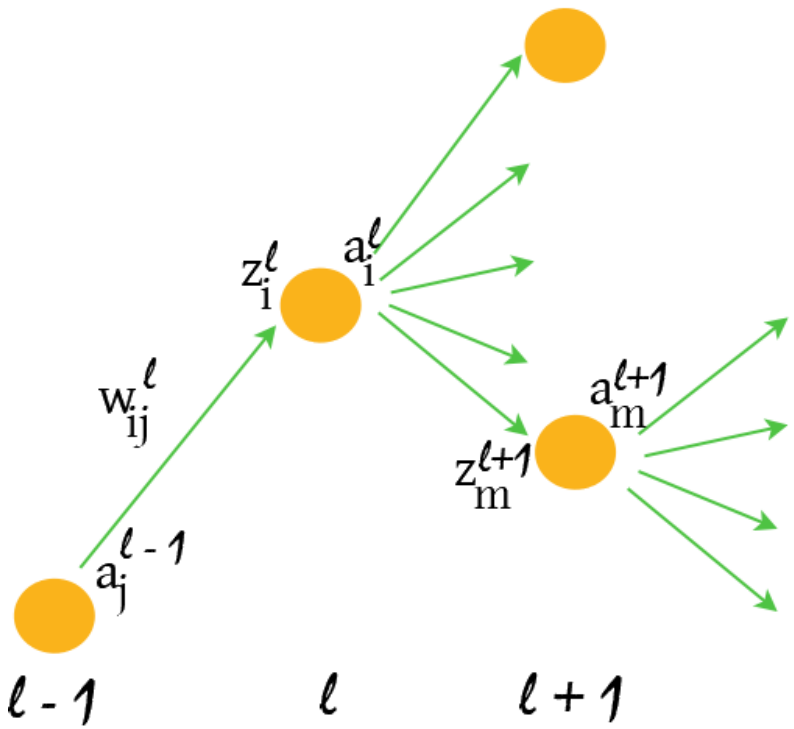
Generalized setting:

$$\frac{\partial e}{\partial w_{ij}^\ell} = \frac{\partial e}{\partial z_i^\ell} a_j^{\ell-1}$$

$$\frac{\partial e}{\partial z_i^\ell} = \frac{\partial e}{\partial a_i^\ell} \phi'(z_i^\ell)$$

$$\frac{\partial e}{\partial a_i^\ell} = \sum_{m=1}^{N_{\ell+1}} \frac{\partial e}{\partial z_m^{\ell+1}} w_{mi}^{\ell+1}$$

GD/SGD for MLP: Sample-wise Gradient Computation



Generalized setting:

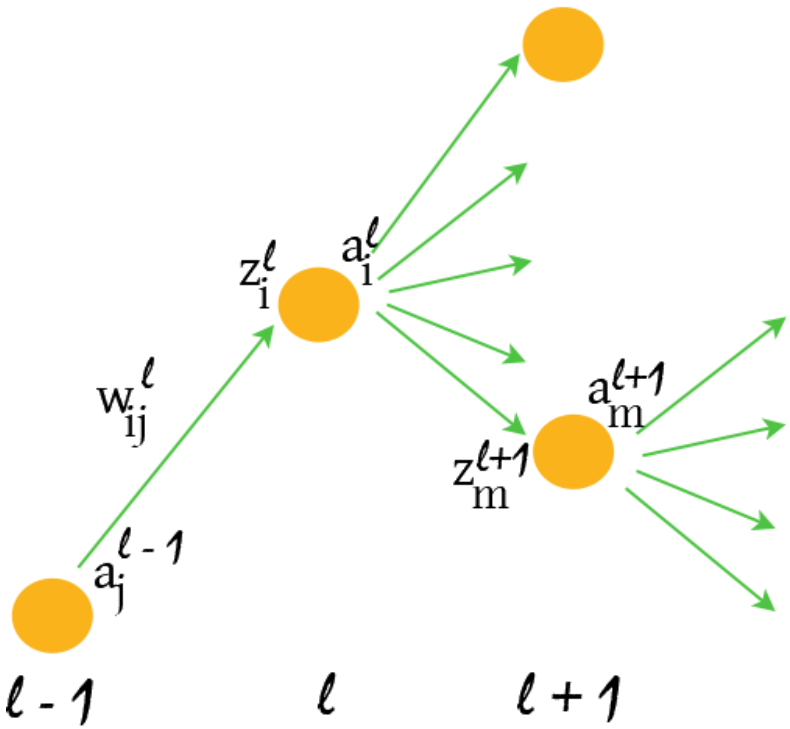
$$\frac{\partial e}{\partial w_{ij}^\ell} = \frac{\partial e}{\partial z_i^\ell} a_j^{\ell-1}$$

$$\frac{\partial e}{\partial z_i^\ell} = \frac{\partial e}{\partial a_i^\ell} \phi'(z_i^\ell)$$

$$\frac{\partial e}{\partial a_i^\ell} = \sum_{m=1}^{N_{\ell+1}} \frac{\partial e}{\partial z_m^{\ell+1}} w_{mi}^{\ell+1}$$

$$= \sum_{m=1}^{N_{\ell+1}} \frac{\partial e}{\partial a_m^{\ell+1}} \phi'(z_m^{\ell+1}) w_{mi}^{\ell+1}$$

GD/SGD for MLP: Sample-wise Gradient Computation



Generalized setting:

$$\frac{\partial e}{\partial w_{ij}^\ell} = \frac{\partial e}{\partial z_i^\ell} a_j^{\ell-1}$$

$$\frac{\partial e}{\partial z_i^\ell} = \frac{\partial e}{\partial a_i^\ell} \phi'(z_i^\ell)$$

$$\frac{\partial e}{\partial a_i^\ell} = \sum_{m=1}^{N_{\ell+1}} \frac{\partial e}{\partial z_m^{\ell+1}} w_{mi}^{\ell+1}$$

$$= \sum_{m=1}^{N_{\ell+1}} \frac{\partial e}{\partial a_m^{\ell+1}} \phi'(z_m^{\ell+1}) w_{mi}^{\ell+1}$$

$$= \begin{bmatrix} \phi'(z_1^{\ell+1}) w_{11}^{\ell+1} & \dots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}1}^{\ell+1} \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell+1}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_{\ell+1}}^{\ell+1}} \end{bmatrix}$$

GD/SGD for MLP: Sample-wise Gradient Computation

Generalized setting:

$$\frac{\partial e}{\partial w_{ij}^{\ell}} = \frac{\partial e}{\partial z_i^{\ell}} a_j^{\ell-1}$$

$$\frac{\partial e}{\partial z_i^{\ell}} = \frac{\partial e}{\partial a_i^{\ell}} \phi'(z_i^{\ell})$$

$$\begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_\ell}^{\ell}} \end{bmatrix} = \begin{bmatrix} \phi'(z_1^{\ell+1}) w_{11}^{\ell+1} & \cdots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}1}^{\ell+1} \\ \vdots & \cdots & \vdots \\ \phi'(z_1^{\ell+1}) w_{1N_\ell}^{\ell+1} & \cdots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}N_\ell}^{\ell+1} \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell+1}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_{\ell+1}}^{\ell+1}} \end{bmatrix}$$

GD/SGD for MLP: Sample-wise Gradient Computation

Generalized setting:

$$\frac{\partial e}{\partial w_{ij}^{\ell}} = \frac{\partial e}{\partial z_i^{\ell}} a_j^{\ell-1}$$

$$\frac{\partial e}{\partial z_i^{\ell}} = \frac{\partial e}{\partial a_i^{\ell}} \phi'(z_i^{\ell})$$

$$\begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_\ell}^{\ell}} \end{bmatrix} = \begin{bmatrix} \phi'(z_1^{\ell+1}) w_{11}^{\ell+1} & \cdots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}1}^{\ell+1} \\ \vdots & \cdots & \vdots \\ \phi'(z_1^{\ell+1}) w_{1N_\ell}^{\ell+1} & \cdots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}N_\ell}^{\ell+1} \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell+1}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_{\ell+1}}^{\ell+1}} \end{bmatrix}$$

$$\implies \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_\ell}^{\ell}} \end{bmatrix} = \begin{bmatrix} w_{11}^{\ell+1} & \cdots & w_{N_{\ell+1}1}^{\ell+1} \\ \vdots & \cdots & \vdots \\ w_{1N_\ell}^{\ell+1} & \cdots & w_{N_{\ell+1}N_\ell}^{\ell+1} \end{bmatrix} \begin{bmatrix} \phi'(z_1^{\ell+1}) & & \\ & \ddots & \\ & & \phi'(z_{N_{\ell+1}}^{\ell+1}) \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell+1}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_{\ell+1}}^{\ell+1}} \end{bmatrix}$$

GD/SGD for MLP: Sample-wise Gradient Computation

Generalized setting:

$$\frac{\partial e}{\partial w_{ij}^{\ell}} = \frac{\partial e}{\partial z_i^{\ell}} a_j^{\ell-1}$$

$$\frac{\partial e}{\partial z_i^{\ell}} = \frac{\partial e}{\partial a_i^{\ell}} \phi'(z_i^{\ell})$$

$$\begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_\ell}^{\ell}} \end{bmatrix} = \begin{bmatrix} \phi'(z_1^{\ell+1}) w_{11}^{\ell+1} & \cdots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}1}^{\ell+1} \\ \vdots & \cdots & \vdots \\ \phi'(z_1^{\ell+1}) w_{1N_\ell}^{\ell+1} & \cdots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}N_\ell}^{\ell+1} \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell+1}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_{\ell+1}}^{\ell+1}} \end{bmatrix}$$

$$\implies \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_\ell}^{\ell}} \end{bmatrix} = \begin{bmatrix} w_{11}^{\ell+1} & \cdots & w_{N_{\ell+1}1}^{\ell+1} \\ \vdots & \cdots & \vdots \\ w_{1N_\ell}^{\ell+1} & \cdots & w_{N_{\ell+1}N_\ell}^{\ell+1} \end{bmatrix} \begin{bmatrix} \phi'(z_1^{\ell+1}) & & \\ & \ddots & \\ & & \phi'(z_{N_{\ell+1}}^{\ell+1}) \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell+1}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_{\ell+1}}^{\ell+1}} \end{bmatrix}$$

$$\delta^{\ell} = (W^{\ell+1})^{\top} \text{Diag}(\phi^{\ell+1'}) \delta^{\ell+1}$$

GD/SGD for MLP: Sample-wise Gradient Computation

Generalized setting:

$$\frac{\partial e}{\partial w_{ij}^{\ell}} = \frac{\partial e}{\partial z_i^{\ell}} a_j^{\ell-1}$$

$$\frac{\partial e}{\partial z_i^{\ell}} = \frac{\partial e}{\partial a_i^{\ell}} \phi'(z_i^{\ell})$$

$$\begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_\ell}^{\ell}} \end{bmatrix} = \begin{bmatrix} \phi'(z_1^{\ell+1}) w_{11}^{\ell+1} & \cdots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}1}^{\ell+1} \\ \vdots & \cdots & \vdots \\ \phi'(z_1^{\ell+1}) w_{1N_\ell}^{\ell+1} & \cdots & \phi'(z_{N_{\ell+1}}^{\ell+1}) w_{N_{\ell+1}N_\ell}^{\ell+1} \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell+1}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_{\ell+1}}^{\ell+1}} \end{bmatrix}$$

$$\implies \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_\ell}^{\ell}} \end{bmatrix} = \begin{bmatrix} w_{11}^{\ell+1} & \cdots & w_{N_{\ell+1}1}^{\ell+1} \\ \vdots & \cdots & \vdots \\ w_{1N_\ell}^{\ell+1} & \cdots & w_{N_{\ell+1}N_\ell}^{\ell+1} \end{bmatrix} \begin{bmatrix} \phi'(z_1^{\ell+1}) & & \\ & \ddots & \\ & & \phi'(z_{N_{\ell+1}}^{\ell+1}) \end{bmatrix} \begin{bmatrix} \frac{\partial e}{\partial a_1^{\ell+1}} \\ \vdots \\ \frac{\partial e}{\partial a_{N_{\ell+1}}^{\ell+1}} \end{bmatrix}$$

$$\delta^{\ell} = (W^{\ell+1})^{\top} \text{Diag}(\phi^{\ell+1'}) \delta^{\ell+1} = V^{\ell+1} \delta^{\ell+1}$$