Recitation 2: Computing Derivatives

Notation and Conventions

- We will refer to the derivative of scalar L with respect to x as $\nabla_x L$
  - Regardless of whether the derivative is a scalar, vector, matrix or tensor

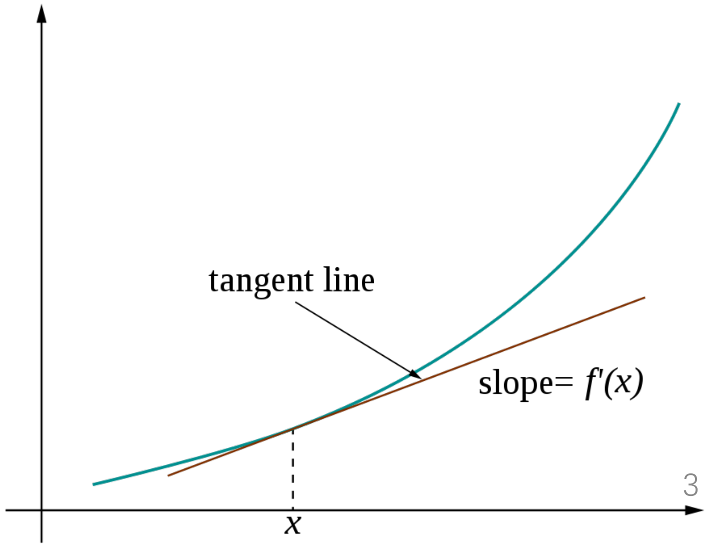
- The derivative of a scalar L w.r.t an $N \times 1$ column vector x is a $1 \times N$ row vector $\nabla_x L$
- The derivative of a scalar L w.r.t an $N \times M$ matrix X is an $M \times N$ matrix $\nabla_X L$
  - Remember our gradient update rule : $W = W - \eta \nabla_W L^T$

- The derivative of an $N \times 1$ vector Y w.r.t an $M \times 1$ vector X is an $N \times M$ matrix $J_X(Y)$
  - The Jacobian

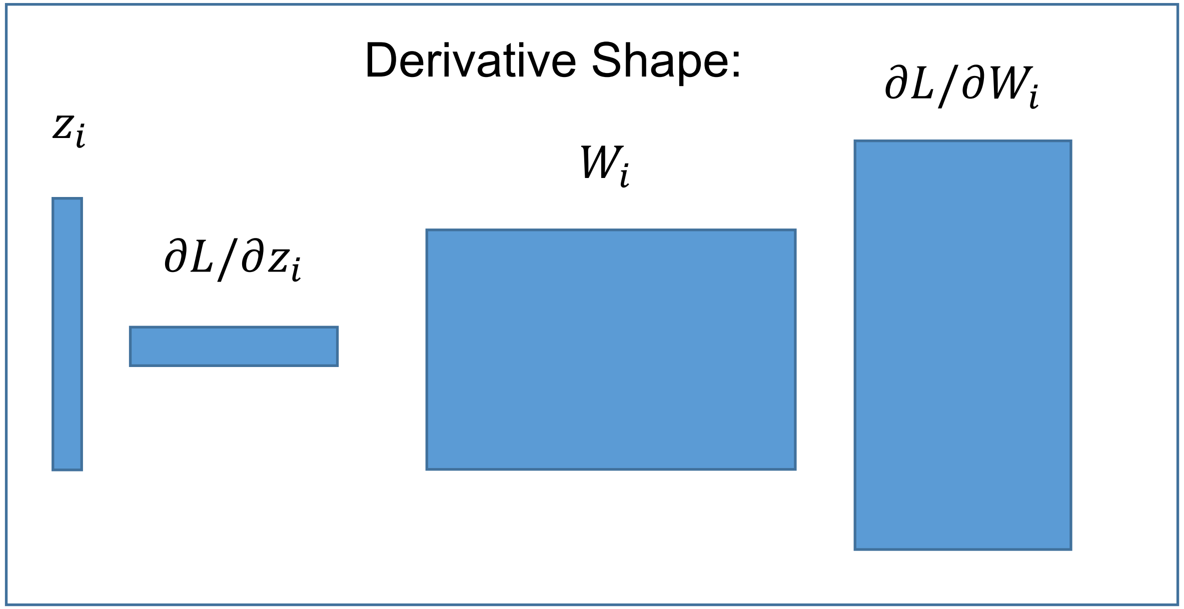
Definition of Derivative

1. Math Definition: $\frac{dy}{dx} = \lim_{\Delta_x \to 0} \frac{\Delta y}{\Delta x}$
2. Intuition:
    - Question: If I increase x by a tiny bit, how much will the overall $f(x)$ increase?
    - Answer: This tiny change will result in $f'(x)$ derivative value change

- Geometrics: The derivative of f w.r.t. *x* at x_0 is the slope of the tangent line to the graph of f at x_0



Computing Derivatives



Notice: the shape of the derivative for any variable will be transposed with respect to that variable

Rule 1(a): Scalar Multiplication

$$z = Wx$$

- All terms are scalars
- $\frac{\partial L}{\partial z}$ is known

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} W$$

$$\frac{\partial L}{\partial W} = x \frac{\partial L}{\partial z}$$

Rule 2(a): Scalar Addition

$$z = x + y$$
$$L = f(z)$$

- All terms are scalars
- $\frac{\partial L}{\partial z}$ is known

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x} = \frac{\partial L}{\partial z}$$
$$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y} = \frac{\partial L}{\partial z}$$

Rule 3(a): Scalar Chain Rule

$$z = g(x)$$
$$L = f(z)$$

- x and z are scalars
- $\frac{\partial L}{\partial z}$ is known

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} g'(x)$$

Rule 4(a): The Generalized Chain Rule (Scalar)

$$L = \; f(g_1(x), g_2(x), \ldots, g_n(x))$$

- x is scalar
- $\frac{\partial L}{\partial g_i}$ are know for all i

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial g_1}\frac{\partial g_1}{\partial x} + \frac{\partial L}{\partial g_2}\frac{\partial g_2}{\partial x} + \cdots + \frac{\partial L}{\partial g_n}\frac{\partial g_n}{\partial x}$$

Rule 1(b): Matrix Multiplication

$$z = Wx$$
$$L = f(z)$$

- z is an $N \times 1$ vector
- x is an $M \times 1$ vector
- W is an $N \times M$ matrix
- L is a function of z
- $\nabla_z L$ is known (and is a $1 \times N$ vector)

$$\nabla_x L = (\nabla_z L)W$$

$$\nabla_W L = x(\nabla_z L)$$

Please verify that the dimensions match!

Rule 2(b): Vector Addition

$$z = x + y$$
$$L = f(z)$$

- x, y and z are all $N \times 1$ vectors
- $\nabla_z L$ is known (and is a $1 \times N$ vector)

$$\nabla_x L = \nabla_z L$$

$$\nabla_y L = \nabla_z L$$

Please verify that the dimensions match!

Rule 3(b): Chain Rule (vector)

$$z = g(x)$$
$$L = f(z)$$

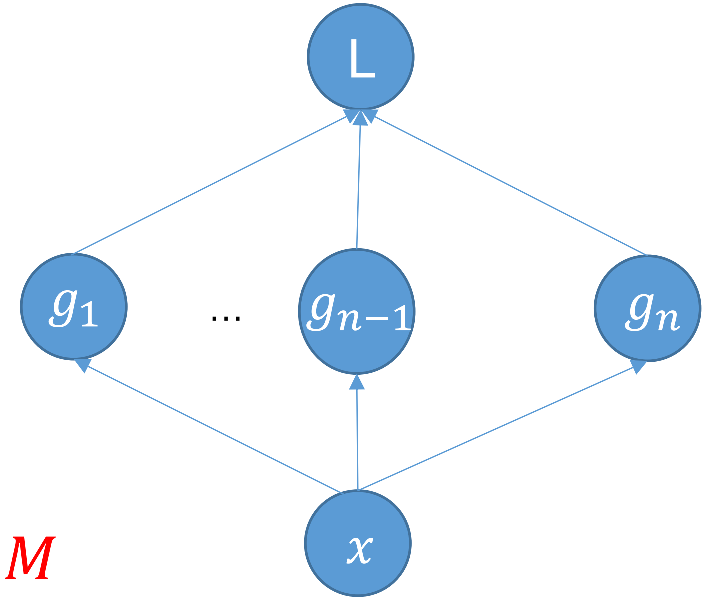
- x and z are $N \times 1$ vectors
- $\nabla_z L$ is known (and is a $1 \times N$ vector)
- $J_x g$ is the *Jacobian* of $g(x)$ with respect to x
  - May be a diagonal matrix

$$\nabla_x L = \nabla_z L \, J_x g$$

Please verify that the dimensions match!

Rule 4(b): The Generalized Chain Rule (vector)

$$L = f(g_1(x), g_2(x), \dots, g_n(x))$$



- x is an $N \times 1$ vector
- The functions g_i output $M \times 1$ vectors for all i
- $\nabla_{g_i} L$ are known for all i (and are $1 \times M$ vectors)
- $J_x g_i$ are $Jacobian\ matrices$ of $g_i(x)$ w.r.t. x of size $M \times N$ matrices.

$$\nabla_x \mathrm{L} = \sum_i \nabla_{g_i} L \, J_x g_i$$

Please verify that the dimensions match!

Rule (5): Element-wise Multiplication

$$z = x \circ y$$
$$L = f(y)$$

- x, y and z are all $N \times 1$ vectors
- "$\circ$" represents component-wise multiplication
- $\nabla_z L$ is known (and is a $1 \times N$ vector)

$$\nabla_x L = (\nabla_z L) \circ y^T$$
$$\nabla_y L = (\nabla_z L) \circ x^T$$

Please verify that the dimensions match!

Rule 6: Element-wise Function

$$z = g(x)$$
$$L = f(z)$$

- x and z are $N \times 1$ vectors
- $\nabla_z L$ is known (and is a $1 \times N$ vector)
- $g(x)$ is actually a vector of *component-wise* functions
  - i.e. $z_i = g(x_i)$
- $g'(x)$ is a row vector consisting of the derivatives of the individual components of $g(x)$ w.r.t individual components of x

$$\nabla_x L = \nabla_z L \circ g'(x)^T$$

Please verify that the dimensions match!

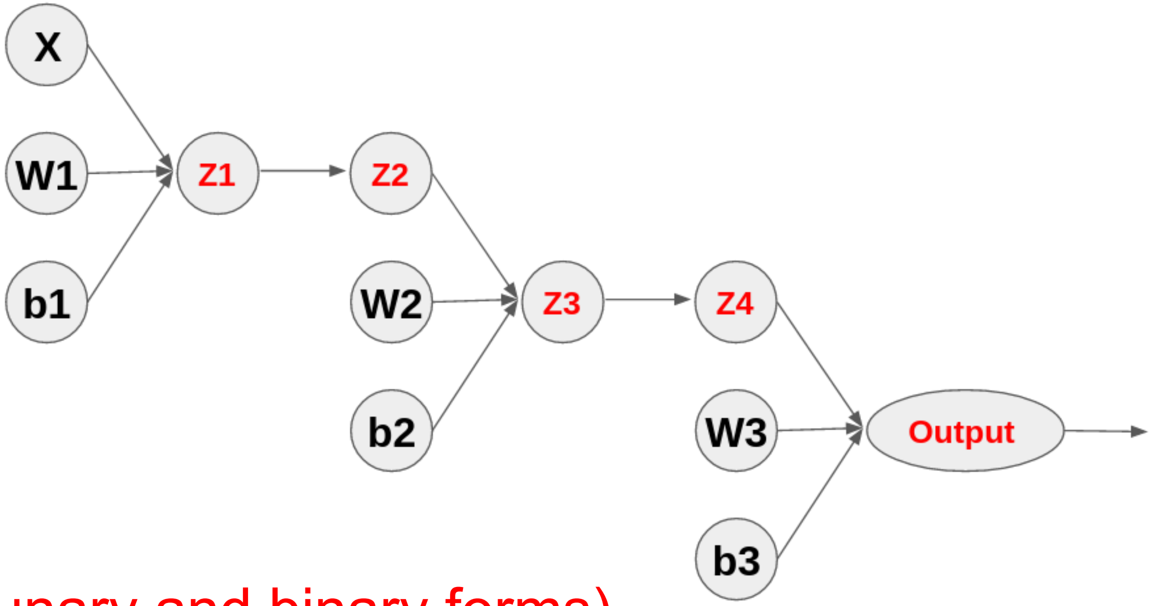
Computing Derivative of Complex Functions

- We now are prepared to compute very complex derivatives
- Given forward computation, the key is to work backward through the simple relations
- Procedure:
  - Express the computation as a series of computations of intermediate values
  - Each computation must comprise either a unary or binary relation
    - Unary relation: RHS has one argument, e.g. $y = g(x)$
    - Binary relation: RHS has two arguments, e.g. $z = x + y$ or $z = xy$

Example 1: MLP Feedfoward Network

- Suppose a MLP network with 2 hidden layers
  Equations of network (in the order in which they are computed sequentially)

1	$z_1 = W_1 x + b_1$
2	$z_2 = relu(z_1)$
3	$z_3 = W_2 z_2 + b_2$
4	$z_4 = relu(z_3)$
5	$output \;\; = W_3 z_4 + b_3$



(Notice that these operations are not in unary and binary forms)

Example 1: MLP Feedfoward Network

Rewrite these in terms of unary and binary operations

1	$z_1 = W_1 x$	1	$z_1 = W_1 x + b_1$
2	$z_2 = z_1 + b_1$	2	$z_2 = relu(z_1)$
3	$z_3 = relu(z_2)$	3	$z_3 = W_2 z_2 + b_2$
4	$z_4 = W_2 z_3$	4	$z_4 = relu(z_3)$
5	$z_5 = z_4 + b_2$	5	$output = W_3 z_4 + b_3$
6	$z_6 = relu(z_5)$		
7	$z_7 = W_3 z_6$		
8	$output = z_7 + b_3$		

Example 1: MLP Backward Network

- Now we will work out way backward
- We assume derivative $\frac{dL}{dOutput}$ of the loss w.r.t. $Output$ is given
- We need to compute $\frac{\partial L}{\partial x}, \frac{\partial L}{\partial W_i}, \frac{\partial L}{\partial b_i}$, which derivative w.r.t. input and parameters within hidden layers

Example 1: MLP Backward Network

1 $\nabla_{z_7} L = \nabla_{output} L$

2 $\nabla_{b_3} L = \nabla_{output} L$

(Recall that for Vector Addition)

$\nabla_x L = \nabla_z L$

$\nabla_y L = \nabla_z L$

1 $z_1 = W_1 x$

2 $z_2 = z_1 + b_1$

3 $z_3 = relu(z_2)$

4 $z_4 = W_2 z_3$

5 $z_5 = z_4 + b_2$

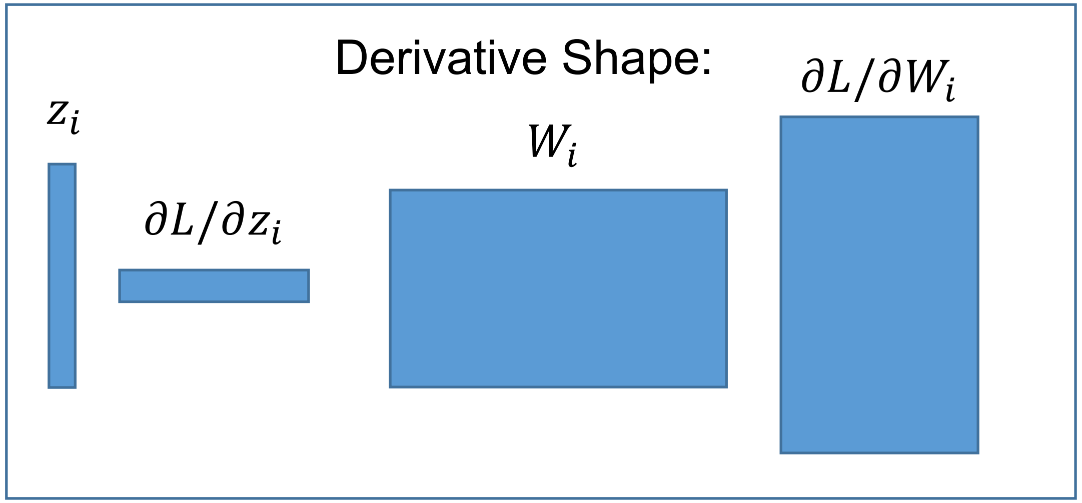
6 $z_6 = relu(z_5)$

7 $z_7 = W_3 z_6$

8 $output = z_7 + b_3$

Example 1: MLP Backward Network

1. $\nabla_{z_7} L = \nabla_{output} L$
2. $\nabla_{b_3} L = \nabla_{output} L$
3. $\nabla_{W_3} L = z_6 \nabla_{z_7} L$
4. $\nabla_{z_6} = \nabla_{z_7} L W_3$



$$
\begin{array}{ll}
1 & z_1 = W_1 x \\
2 & z_2 = z_1 + b_1 \\
3 & z_3 = relu(z_2) \\
4 & z_4 = W_2 z_3 \\
5 & z_5 = z_4 + b_2 \\
6 & z_6 = relu(z_5) \\
7 & z_7 = W_3 z_6 \\
8 & output = z_7 + b_3
\end{array}
$$

Example 1: MLP Backward Network

1. $\nabla_{z_7} L = \nabla_{output} L$
2. $\nabla_{b_3} L = \nabla_{output} L$
3. $\nabla_{W_3} L = z_6 \nabla_{z_7} L$
4. $\nabla_{z_6} L = \nabla_{z_7} L W_3$
5. $\nabla_{z_5} L = \nabla_{z_6} L \circ 1_A(z_5)^T$

$$1_A(z_5) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

1 $z_1 = W_1 x$
2 $z_2 = z_1 + b_1$
3 $z_3 = relu(z_2)$
4 $z_4 = W_2 z_3$
5 $z_5 = z_4 + b_2$
6 $z_6 = relu(z_5)$
7 $z_7 = W_3 z_6$
8 $output = z_7 + b_3$

Recall element-wise function, where $g(x)$ is element-wise funtion

$$\nabla_x L = \nabla_z L \circ g'(x)^T$$

Example 1: MLP Backward Network

1. $\nabla_{z_7} L = \nabla_{output} L$
2. $\nabla_{b_3} L = \nabla_{output} L$
3. $\nabla_{W_3} L = z_6 \nabla_{z_7} L$
4. $\nabla_{z_6} L = \nabla_{z_7} L W_3$
5. $\nabla_{z_5} L = \nabla_{z_6} L \circ 1_A(z_5)^T$

$$1_A(z_5) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

6. $\nabla_{z_4} L = \nabla_{z_5} L$
7. $\nabla_{b_2} L = \nabla_{z_5} L$

$1 \quad z_1 = W_1 x$
$2 \quad z_2 = z_1 + b_1$
$3 \quad z_3 = relu(z_2)$
$4 \quad z_4 = W_2 z_3$
$5 \quad z_5 = z_4 + b_2$
$6 \quad z_6 = relu(z_5)$
$7 \quad z_7 = W_3 z_6$
$8 \quad output = z_7 + b_3$

Example 1: MLP Backward Network

1. $\nabla_{z_7} L = \nabla_{output} L$
2. $\nabla_{b_3} L = \nabla_{output} L$
3. $\nabla_{W_3} L = z_6 \nabla_{z_7} L$
4. $\nabla_{z_6} L = \nabla_{z_7} L W_3$
5. $\nabla_{z_5} L = \nabla_{z_6} L \circ 1_A(z_5)^T$

$$1_A(z_5) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

6. $\nabla_{z_4} L = \nabla_{z_5} L$
7. $\nabla_{b_2} L = \nabla_{z_5} L$
8. $\nabla_{W_2} L = z_3 \nabla_{z_4} L$
9. $\nabla_{z_3} L = \nabla_{z_4} L W_2$

1 $z_1 = W_1 x$
2 $z_2 = z_1 + b_1$
3 $z_3 = relu(z_2)$
4 $z_4 = W_2 z_3$
5 $z_5 = z_4 + b_2$
6 $z_6 = relu(z_5)$
7 $z_7 = W_3 z_6$
8 $output = z_7 + b_3$

Example 1: MLP Backward Network

6. $\nabla_{z_4} L = \nabla_{z_5} L$
7. $\nabla_{b_2} L = \nabla_{z_5} L$
8. $\nabla_{W_2} L = z_3 \nabla_{z_4} L$
9. $\nabla_{z_3} L = \nabla_{z_4} L W_2$
10. $\nabla_{z_2} L = \nabla_{z_3} L \circ 1_A(z_5)^T$

$$1_A(z_5) = \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases}$$

1 $z_1 = W_1 x$
2 $z_2 = z_1 + b_1$
3 $z_3 = relu(z_2)$
4 $z_4 = W_2 z_3$
5 $z_5 = z_4 + b_2$
6 $z_6 = relu(z_5)$
7 $z_7 = W_3 z_6$
8 $output = z_7 + b_3$

Example 1: MLP Backward Network

6. $\nabla_{z_4} L = \nabla_{z_5} L$
7. $\nabla_{b_2} L = \nabla_{z_5} L$
8. $\nabla_{W_2} L = z_3 \nabla_{z_4} L$
9. $\nabla_{z_3} L = \nabla_{z_4} L W_2$
10. $\nabla_{z_2} L = \nabla_{z_3} L \circ 1_A(z_5)^T$

$$1_A(z_5) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

11. $\nabla_{b_1} L = \nabla_{z_2} L$
12. $\nabla_{z_1} L = \nabla_{z_2} L$

1 $z_1 = W_1 x$
2 $z_2 = z_1 + b_1$
3 $z_3 = relu(z_2)$
4 $z_4 = W_2 z_3$
5 $z_5 = z_4 + b_2$
6 $z_6 = relu(z_5)$
7 $z_7 = W_3 z_6$
8 $output = z_7 + b_3$

Example 1: MLP Backward Network

6. $\nabla_{z_4} L = \nabla_{z_5} L$
7. $\nabla_{b_2} L = \nabla_{z_5} L$
8. $\nabla_{W_2} L = z_3 \nabla_{z_4} L$
9. $\nabla_{z_3} L = \nabla_{z_4} L W_2$
10. $\nabla_{z_2} L = \nabla_{z_3} L \circ 1_A(z_5)^T$

$$1_A(z_5) = \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases}$$

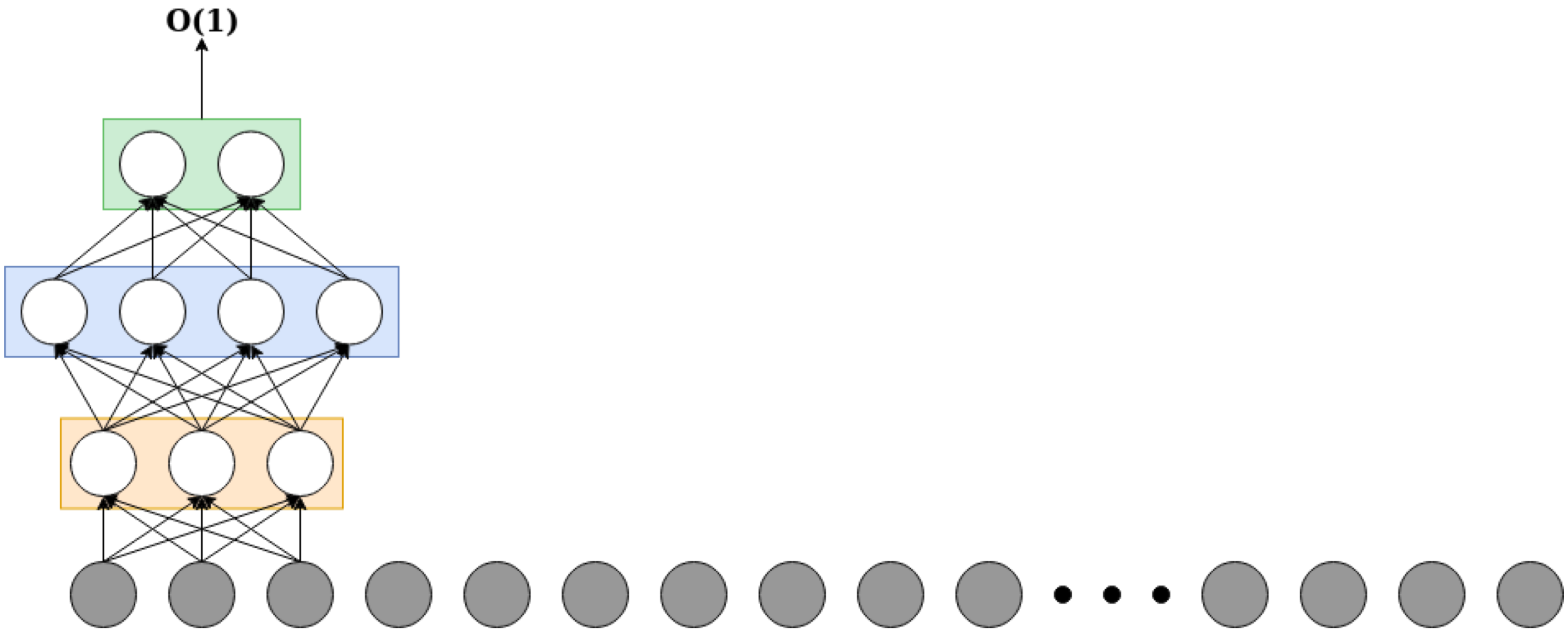
11. $\nabla_{b_1} L = \nabla_{z_2} L$
12. $\nabla_{z_1} L = \nabla_{z_2} L$
13. $\nabla_{W_1} L = x \nabla_{z_1} L$
14. $\nabla_x L = \nabla_{z_1} L W_1$

1 $z_1 = W_1 x$
2 $z_2 = z_1 + b_1$
3 $z_3 = relu(z_2)$
4 $z_4 = W_2 z_3$
5 $z_5 = z_4 + b_2$
6 $z_6 = relu(z_5)$
7 $z_7 = W_3 z_6$
8 $output = z_7 + b_3$

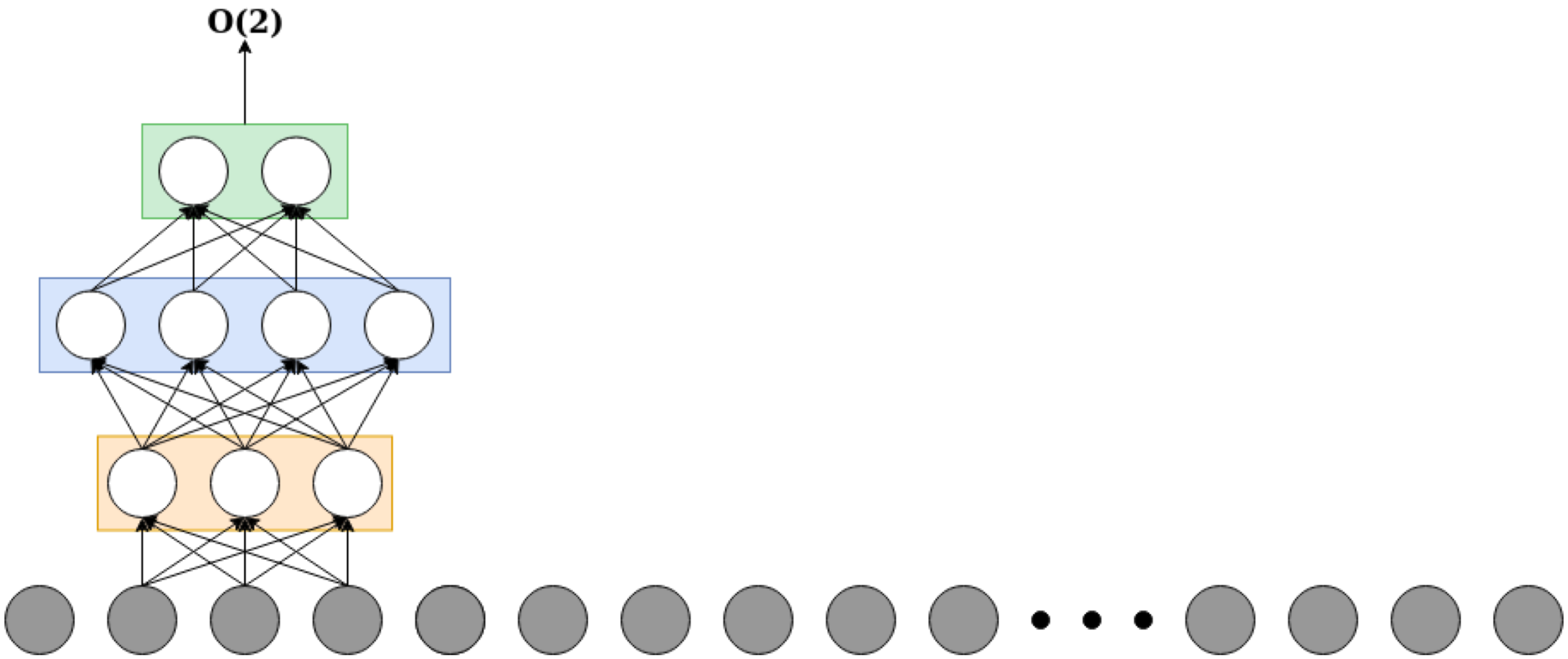
Example 2: Scanning with an MLP

- **X** is a T x 1 vector
- The MLP takes an input vector x(t) = **X**[t : t + N, :] of size N x 1 at each step t
- O(t) is the output of the MLP at step t

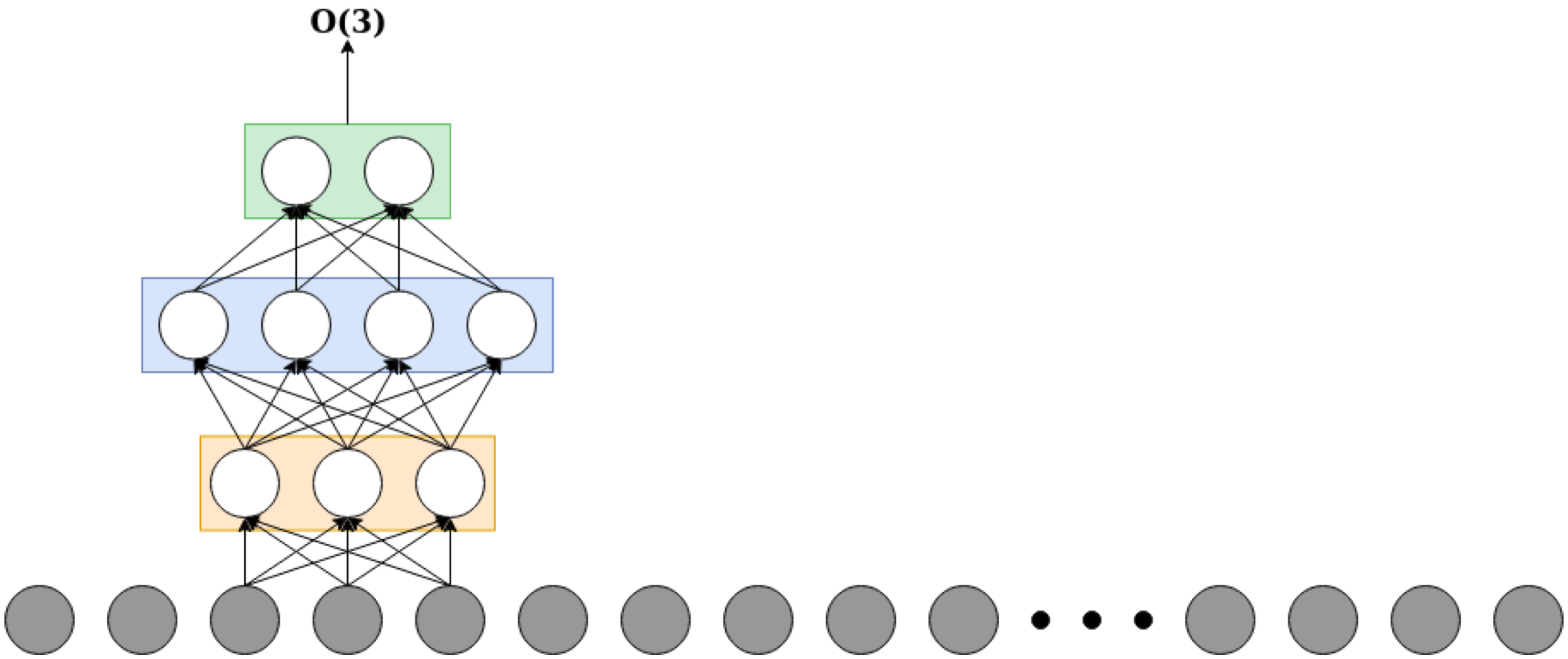
Example 2: Scanning with an MLP



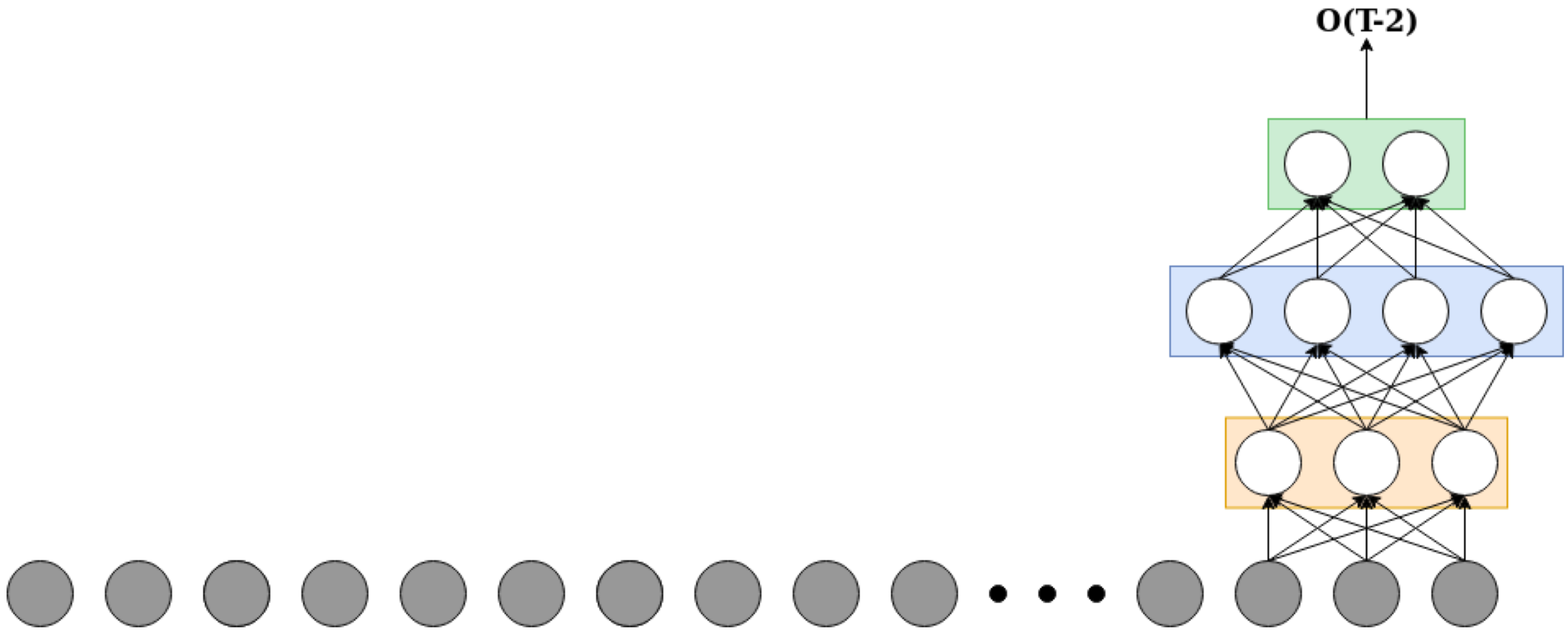
Example 2: Scanning with an MLP



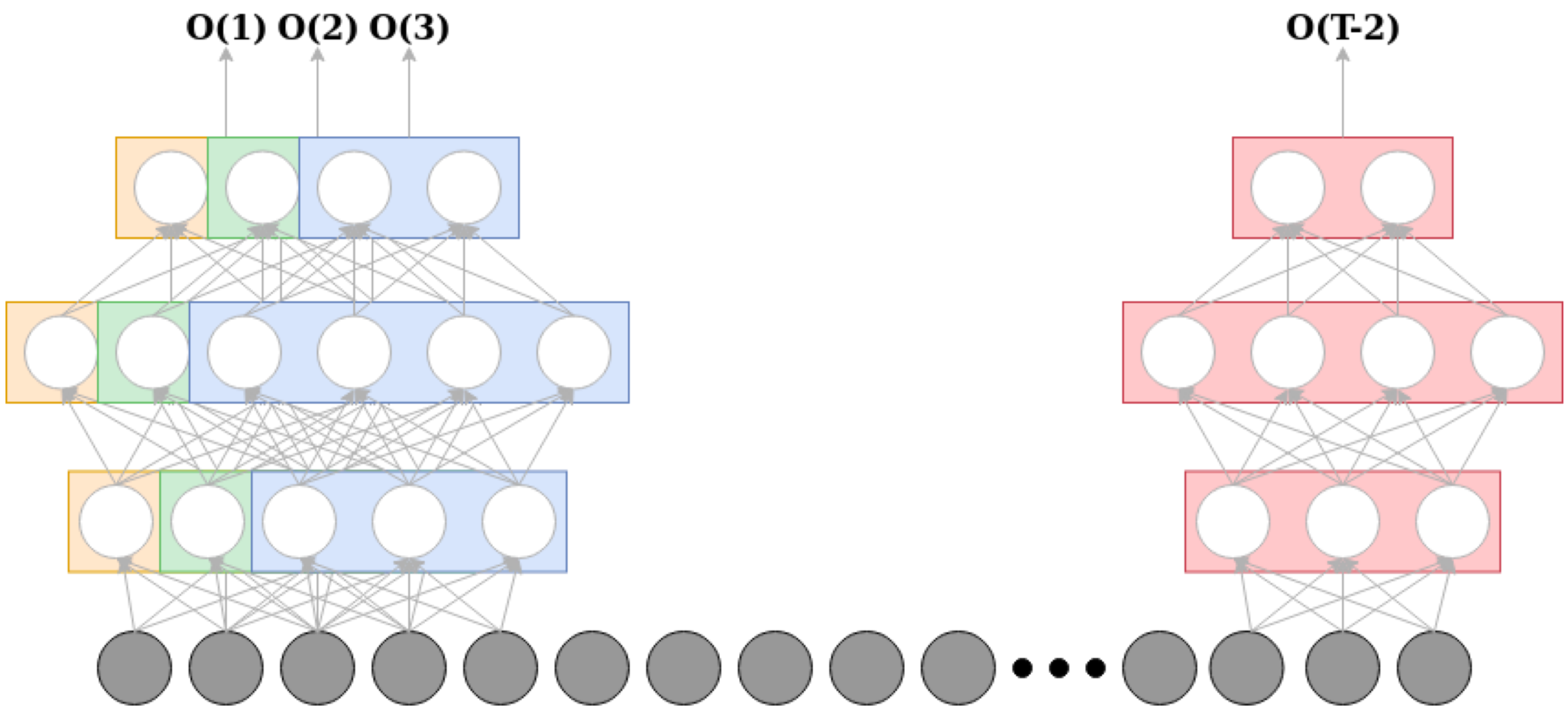
Example 2: Scanning with an MLP



Example 2: Scanning with an MLP



Example 2: Scanning with an MLP



Example 2: Scanning with an MLP (forward)

- **X** is a T x 1 vector
- The MLP takes an input vector x(t) = **X**[t : t + N, :] of size N x 1 at each step t
- O(t) is the output of the MLP at step t
- L = f(O(1), O(2), …, O(T-N+1))
- Forward equations of the network at step t:

1. $z_1(t) = W_1 x(t) + b_1$
2. $z_2(t) = relu\big(z_1(t)\big)$
3. $z_3(t) = W_2 z_2 + b_2$
4. $z_4(t) = relu(z_3(t))$
5. $O(t) = W_3 z_4(t) + b_3$

Example 2: Scanning with an MLP (forward)

Rewrite these in terms of unary and binary operations

1. $z_1(t) = W_1 x(t)$
2. $z_2(t) = z_1(t) + b_1$
3. $z_3(t) = relu\big(z_2(t)\big)$
4. $z_4(t) = W_2 z_3$
5. $z_5(t) = z_4 + b_2$
6. $z_6(t) = relu\big(z_5(t)\big)$
7. $z_7(t) = W_3 z_6(t)$
8. $O(t) = z_7(t) + b_3$

1. $z_1(t) = \mathrm{W}_1 x(t) + b_1$
2. $z_2(t) = relu\big(z_1(t)\big)$
3. $z_3(t) = \mathrm{W}_2 z_2 + b_2$
4. $z_4(t) = relu\big(z_3(t)\big)$
5. $O(t) = W_3 z_4(t) + b_3$

Example 2: Scanning with an MLP (backward)

- Let's now work our way backward
- We assume derivative $\frac{dL}{dO(t)}$ of the loss w.r.t. $O(t)$ is given for t=1,...,T-N+1
- We need to compute $\frac{dL}{dX}, \frac{dL}{dW_i}, \frac{dL}{db_i}$ the derivatives of the loss w.r.t. the inputs and the network parameters

Example 2: Scanning with an MLP (backward)

Calculating the derivatives for t = 1:

1. $\nabla_{z_{7(t)}} L = \nabla_{O(t)} L$
2. $\nabla_{b_3} L = \nabla_{O(t)} L$
3. $\nabla_{W_3} L = z_6(t) \nabla_{z_{7(t)}} L$
4. $\nabla_{z_6(t)} L = \nabla_{z_7(t)} L W_3$
5. $\nabla_{z_5(t)} L = \nabla_{z_6(t)} L \circ 1_A(z_5(t))^T$

$$1_A(z_5(t)) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

6. $\nabla_{z_4(t)} L = \nabla_{z_5(t)} L$
7. $\nabla_{b_2} L = \nabla_{z_5(t)} L$
8. $\nabla_{W_2} L = z_3(t) \nabla_{z_4(t)} L$
9. $\nabla_{z_3(t)} L = \nabla_{z_4(t)} L W_2$
10. $\nabla_{z_2(t)} L = \nabla_{z_3(t)} L \circ 1_A(z_5(t))^T$

$$1_A(z_5(t)) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

11. $\nabla_{b_1} L = \nabla_{z_2(t)} L$
12. $\nabla_{z_1(t)} L = \nabla_{z_2(t)} L$
13. $\nabla_{W_1} L = x(t) \nabla_{z_1(t)} L$
14. $\nabla_{x(t)} L = \nabla_{z_1(t)} L W_1$
15. $\nabla_X L[:, 1: N + 1] = \nabla_{x(t)} L$

Example 2: Scanning with an MLP (backward)

Calculating the derivatives for t > 1:

1. $\nabla_{z_{7(t)}} L = \nabla_{O(t)} L$
2. $\nabla_{b_3} L += \nabla_{O(t)} L$
3. $\nabla_{W_3} L += z_6(t) \nabla_{z_{7(t)}} L$
4. $\nabla_{z_6(t)} L = \nabla_{z_7(t)} L W_3$
5. $\nabla_{z_5(t)} L = \nabla_{z_6(t)} L \circ 1_A(z_5(t))^T$

$$1_A(z_5(t)) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

6. $\nabla_{z_4(t)} L = \nabla_{z_5(t)} L$
7. $\nabla_{b_2} L += \nabla_{z_5(t)} L$
8. $\nabla_{W_2} L += z_3(t) \nabla_{z_4(t)} L$
9. $\nabla_{z_3(t)} L = \nabla_{z_4(t)} L W_2$
10. $\nabla_{z_2(t)} L = \nabla_{z_3(t)} L \circ 1_A(z_5(t))^T$

$$1_A(z_5(t)) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$$

11. $\nabla_{b_1} L += \nabla_{z_2(t)} L$
12. $\nabla_{z_1(t)} L = \nabla_{z_2(t)} L$
13. $\nabla_{W_1} L += x(t) \nabla_{z_1(t)} L$
14. $\nabla_{x(t)} L = \nabla_{z_1(t)} L W_1$
15. $\nabla_X L[:, t : t + N - 1] += \nabla_{x(t)} L[:, : -1]$
16. $\nabla_X L[:, t + N - 1] = \nabla_{x(t)} L[:, -1]$

When to use "=" vs "+"

- In the forward computation, a variable may be used multiple times to compute other intermediate variables or a sequence of output variables
- During backward computations, the first time the derivative is computed for the variable, the we will use "="
- In subsequent computations we use "+="
- It may be difficult to keep track of when we first compute the derivative for a variable

  When to use "=" vs when to use "+="
- Cheap trick:
  - Initialize all derivatives to 0 during computation
  - Always use "+="
  - You will get the correct answer (why?)