Proof by Examples: Computing Derivatives Can be Trivial

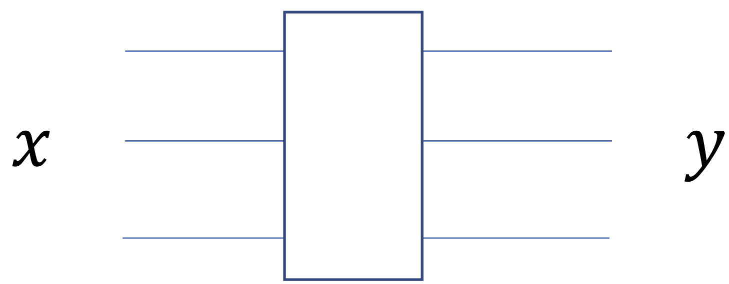
Influence Diagrams

A New (Made Up) Activation in Town

$$y_i = \cos\left(\frac{e^{x_i} \quad \sum_j x_j}{\sum_j \ln(x_j)}\right)$$

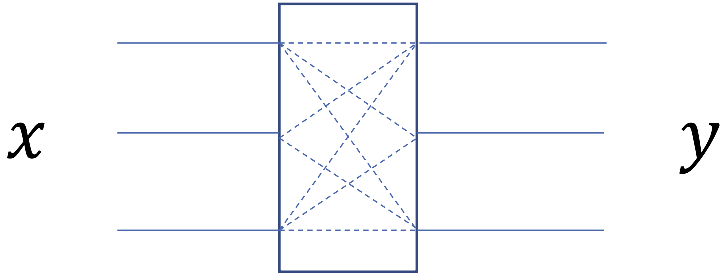
A New (Made Up) Activation in Town

$$y_i = \cos\left(\frac{e^{x_i} \; \sum_j x_j}{\sum_j \ln(x_j)}\right)$$



A New (Made Up) Activation in Town

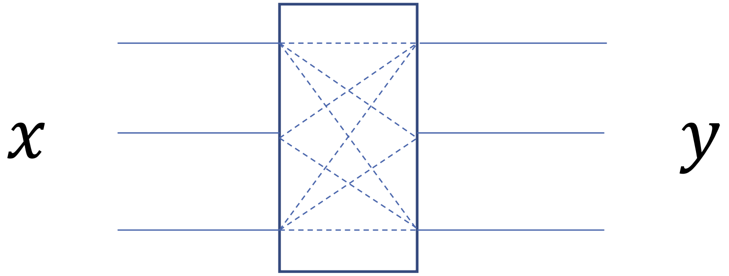
$$y_i = \cos\left(\frac{e^{x_i} \; \sum_j x_j}{\sum_j \ln(x_j)}\right)$$



This is a **vector activation**, as inputs affect multiple outputs

A New (Made Up) Activation in Town

$$y_i = \cos\left(\frac{e^{x_i} \ \sum_j x_j}{\sum_j \ln(x_j)}\right)$$



Let's calculate derivatives

Goal: $\nabla_x L$

A New (Made Up) Activation in Town

$$y_i = \cos\left(\frac{e^{x_i} \quad \sum_j x_j}{\sum_j \ln(x_j)}\right)$$

First we'll break things up so
they're manageable…

A New (Made Up) Activation in Town

$$y_i = \cos(\frac{e^{x_i \sum_j x_j}}{\sum_j \ln(x_j)})$$

$$a = \sum_j x_j$$

$$b = \sum_j \ln(x_j)$$

$$y_i = \cos(\frac{e^{x_i a}}{b})$$

First we'll break things up so they're manageable…

A New (Made Up) Activation in Town

$$a = \sum_j x_j$$

$$b = \sum_j \ln(x_j)$$

$$y_i = \cos(\frac{e^{x_i \quad a}}{b})$$

First we'll break things up so
they're manageable…

A New (Made Up) Activation in Town

$$a = \sum_j x_j$$

$$b = \sum_j \ln(x_j)$$

$$y_i = \cos(\frac{e^{x_i \ a}}{b})$$

Now we'll draw the influence diagram…

A New (Made Up) Activation in Town

$$a = \sum_j x_j$$

$$b = \sum_j \ln(x_j)$$

$$y_i = \cos(\frac{e^{x_i \; a}}{b})$$

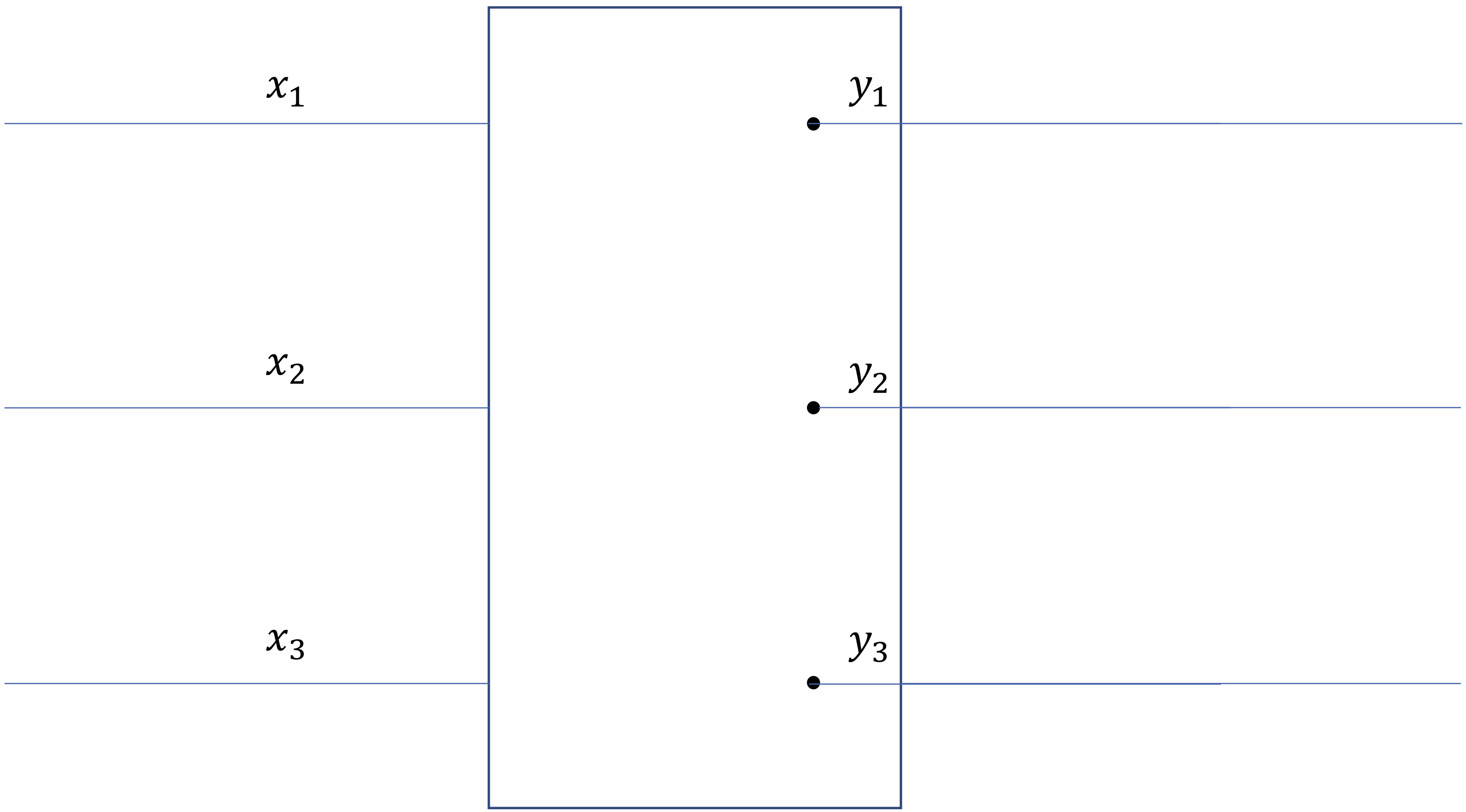
Now we'll draw the influence diagram…

A New (Made Up) Activation in Town

$$a = \sum_j x_j$$

$$b = \sum_j \ln(x_j)$$

$$y_i = \cos\left(\frac{e^{x_i\ a}}{b}\right)$$

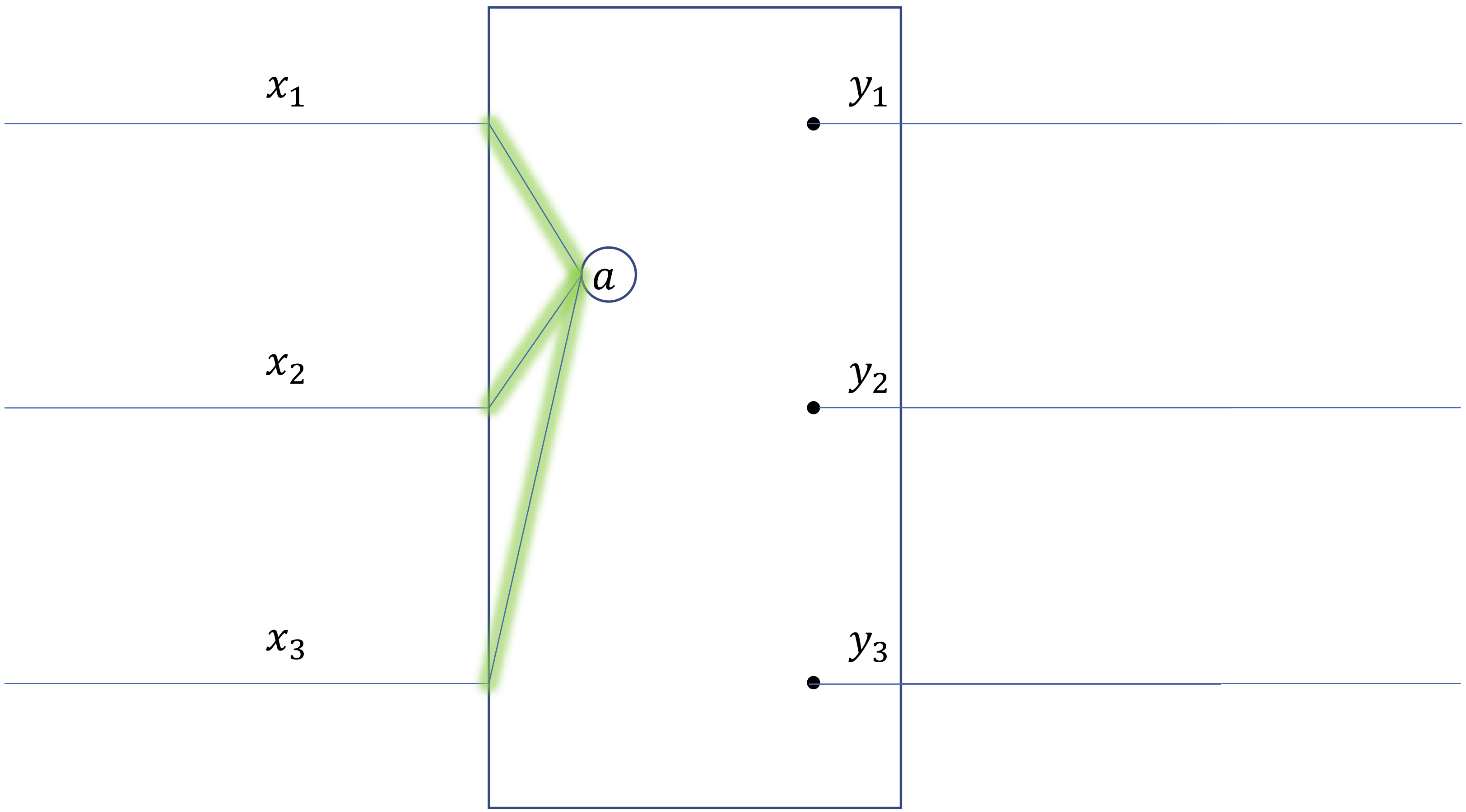


A New (Made Up) Activation in Town

$$a = \sum_j x_j$$

$$b = \sum_j \ln(x_j)$$

$$y_i = \cos\left(\frac{e^{x_i a}}{b}\right)$$

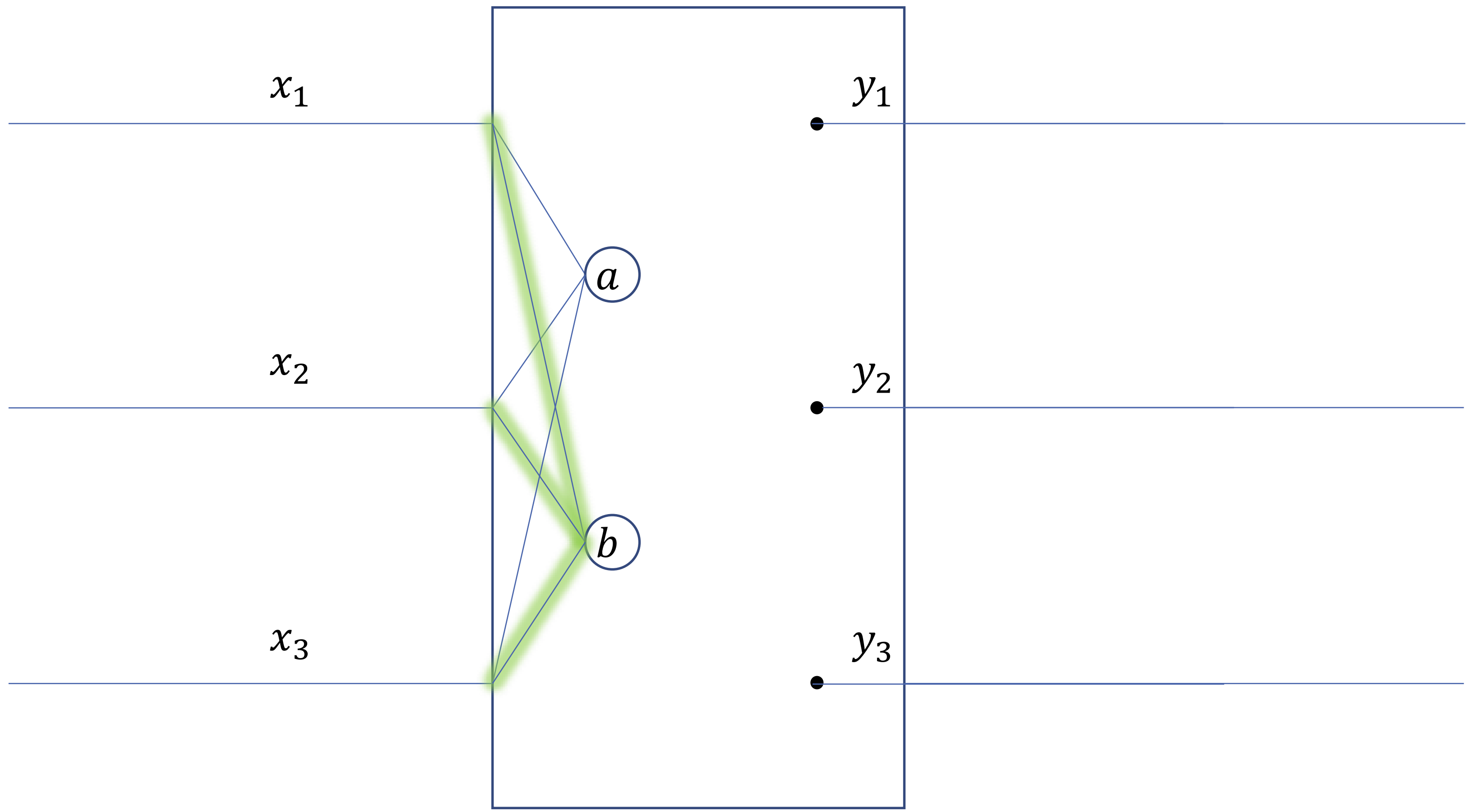


A New (Made Up) Activation in Town

$$a = \sum_j x_j$$

$$b = \sum_j \ln(x_j)$$

$$y_i = \cos(\frac{e^{x_i \ a}}{b})$$

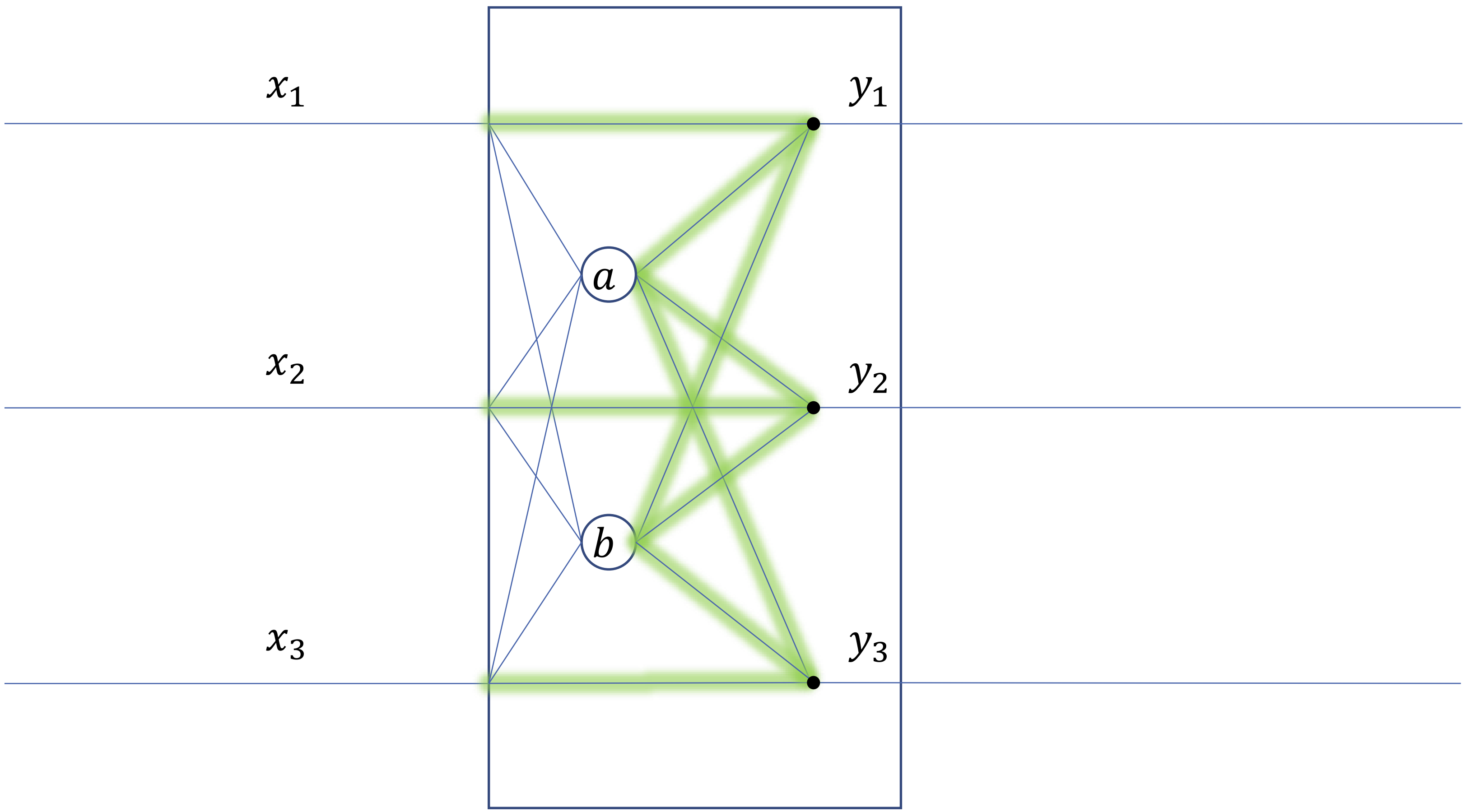


A New (Made Up) Activation in Town

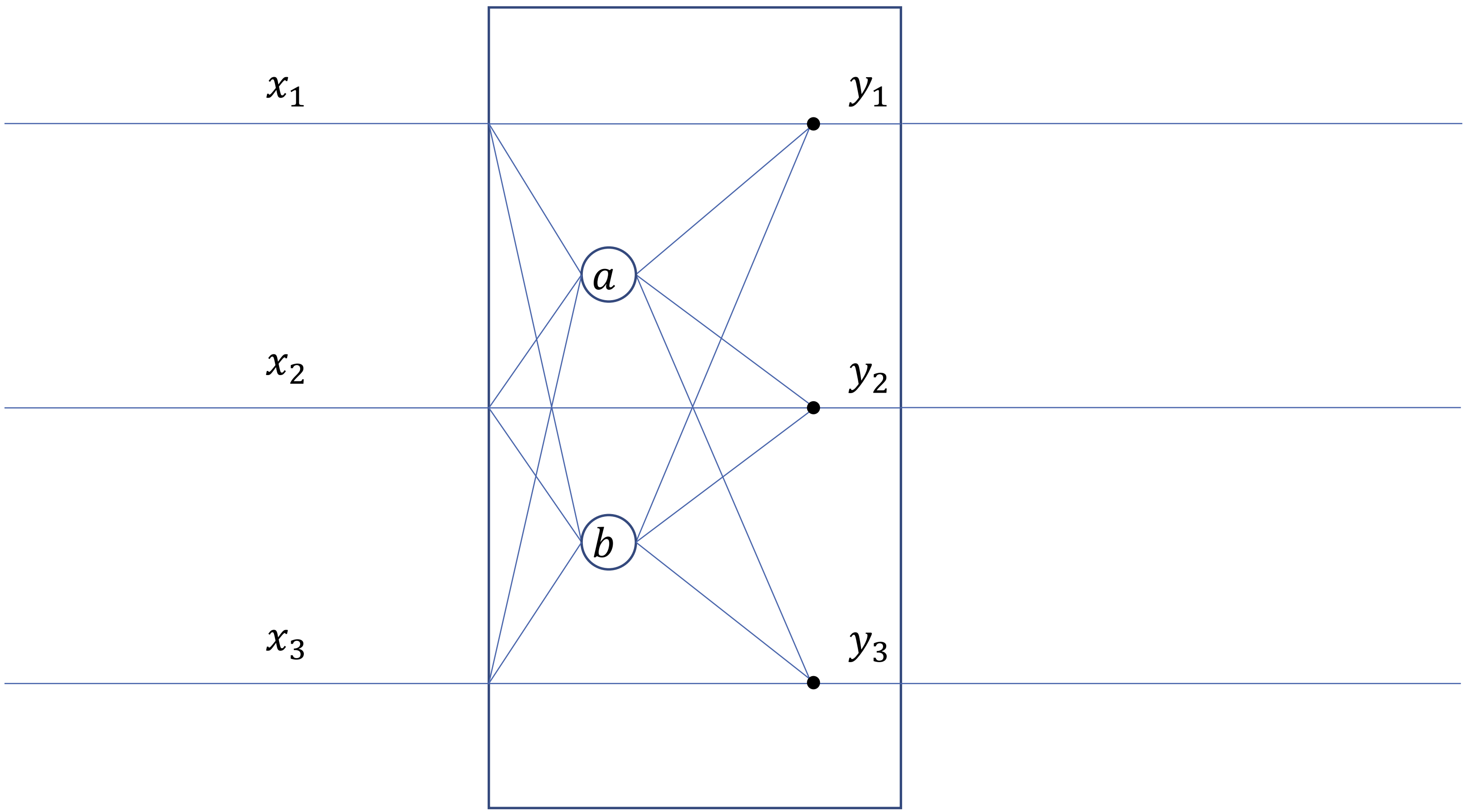
$$a = \sum_j x_j$$

$$b = \sum_j \ln(x_j)$$

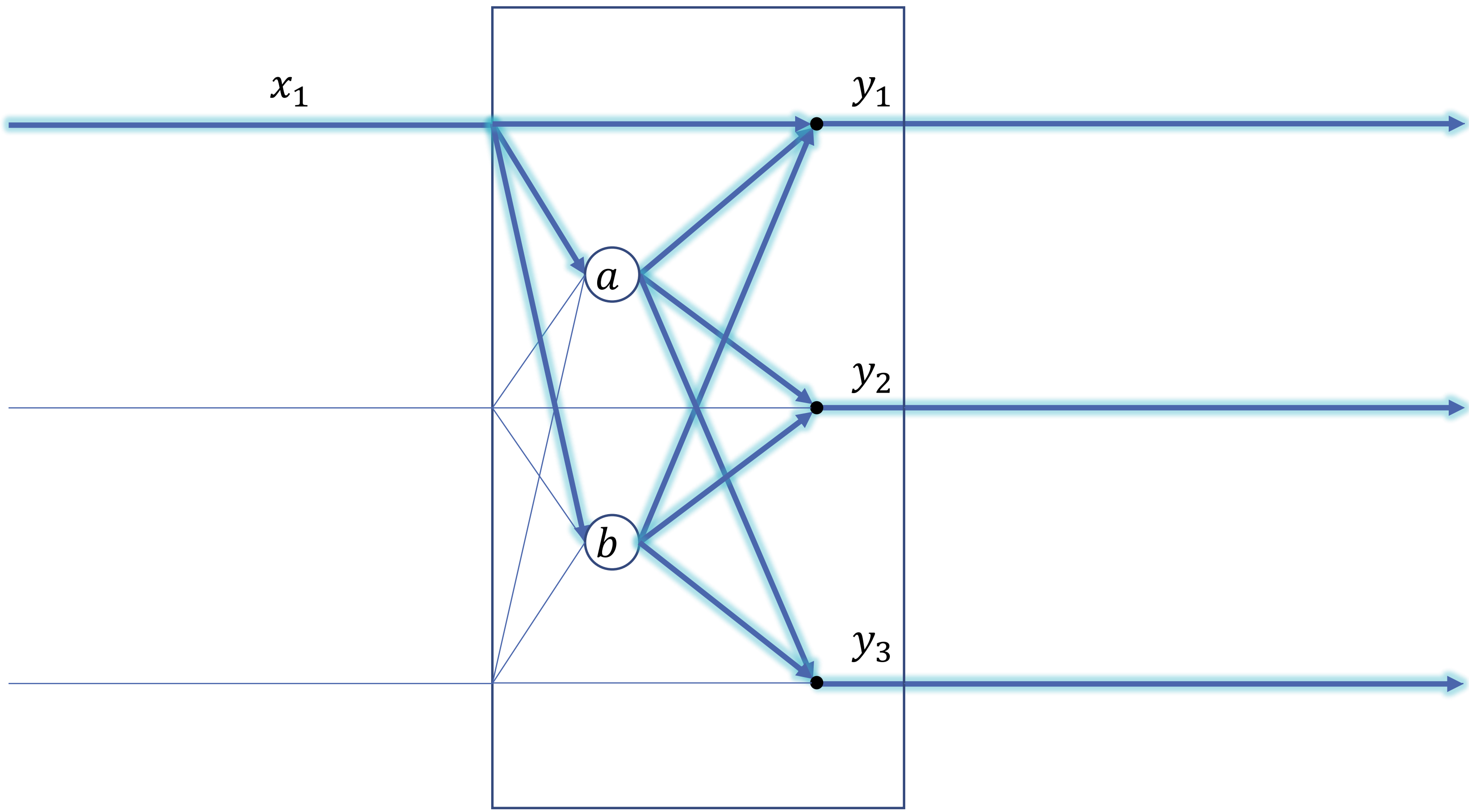
$$y_i = \cos\left(\frac{e^{x_i\ a}}{b}\right)$$



A New (Made Up) Activation in Town

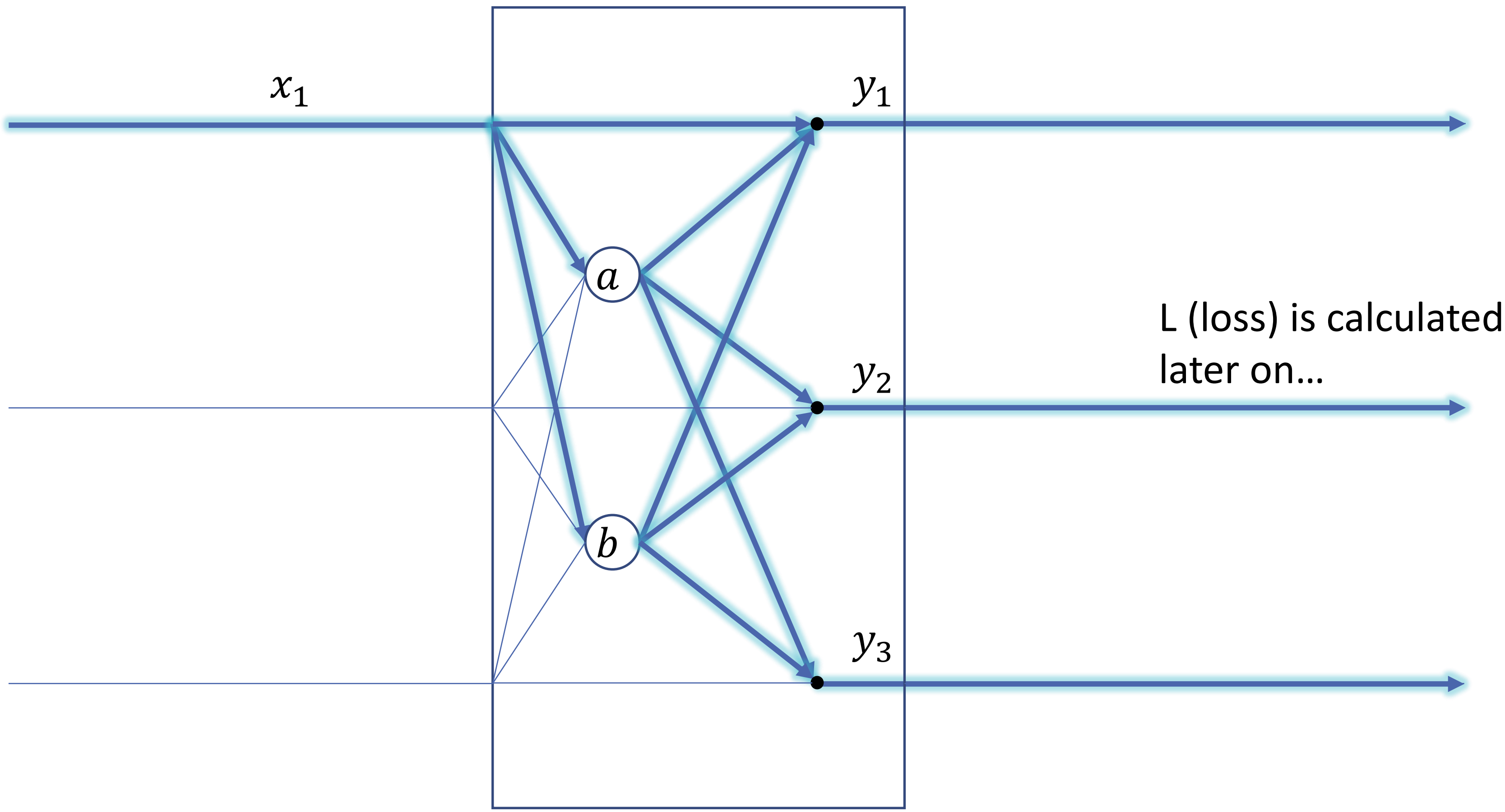


A New (Made Up) Activation in Town



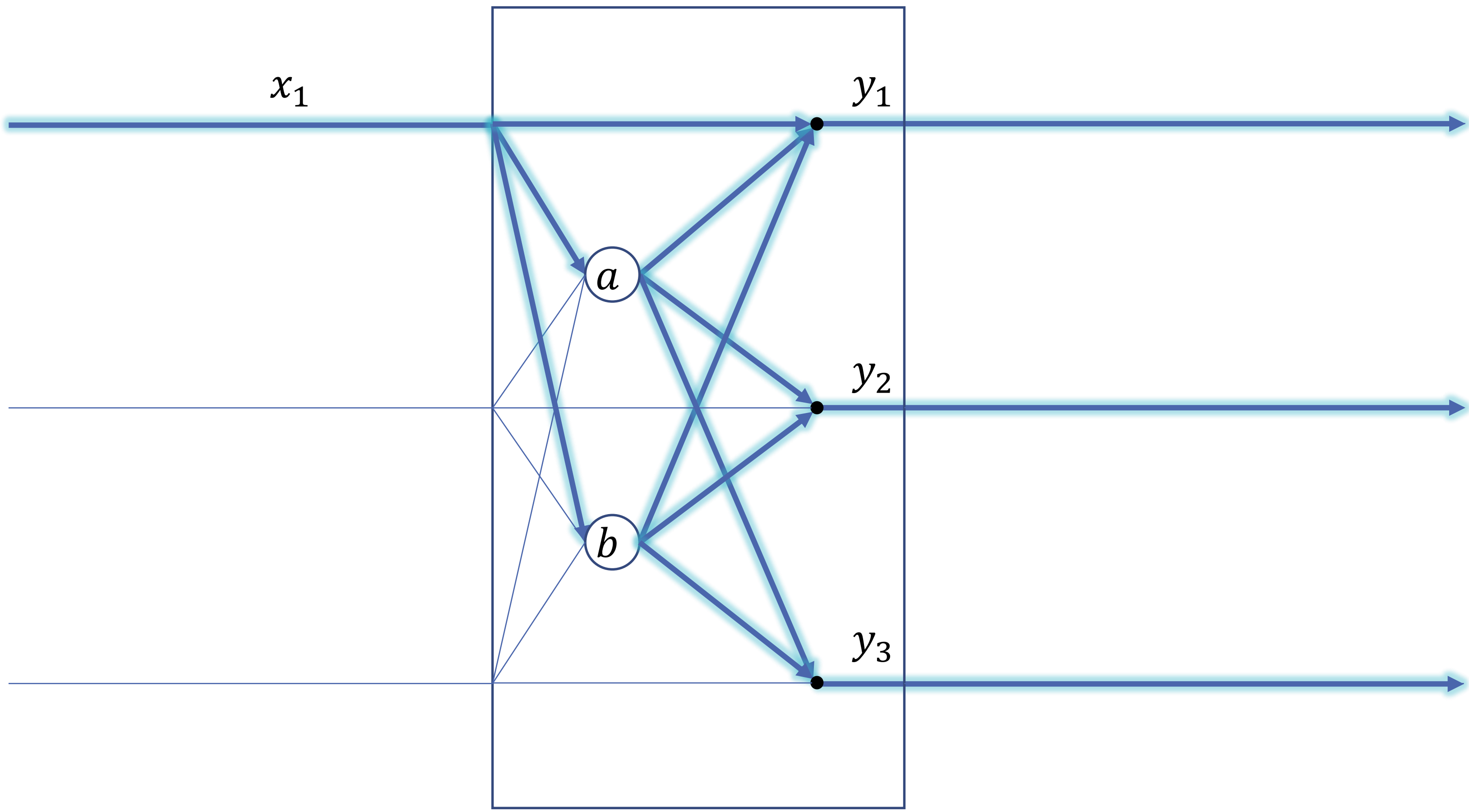
Notice x_1's different paths of influence

A New (Made Up) Activation in Town



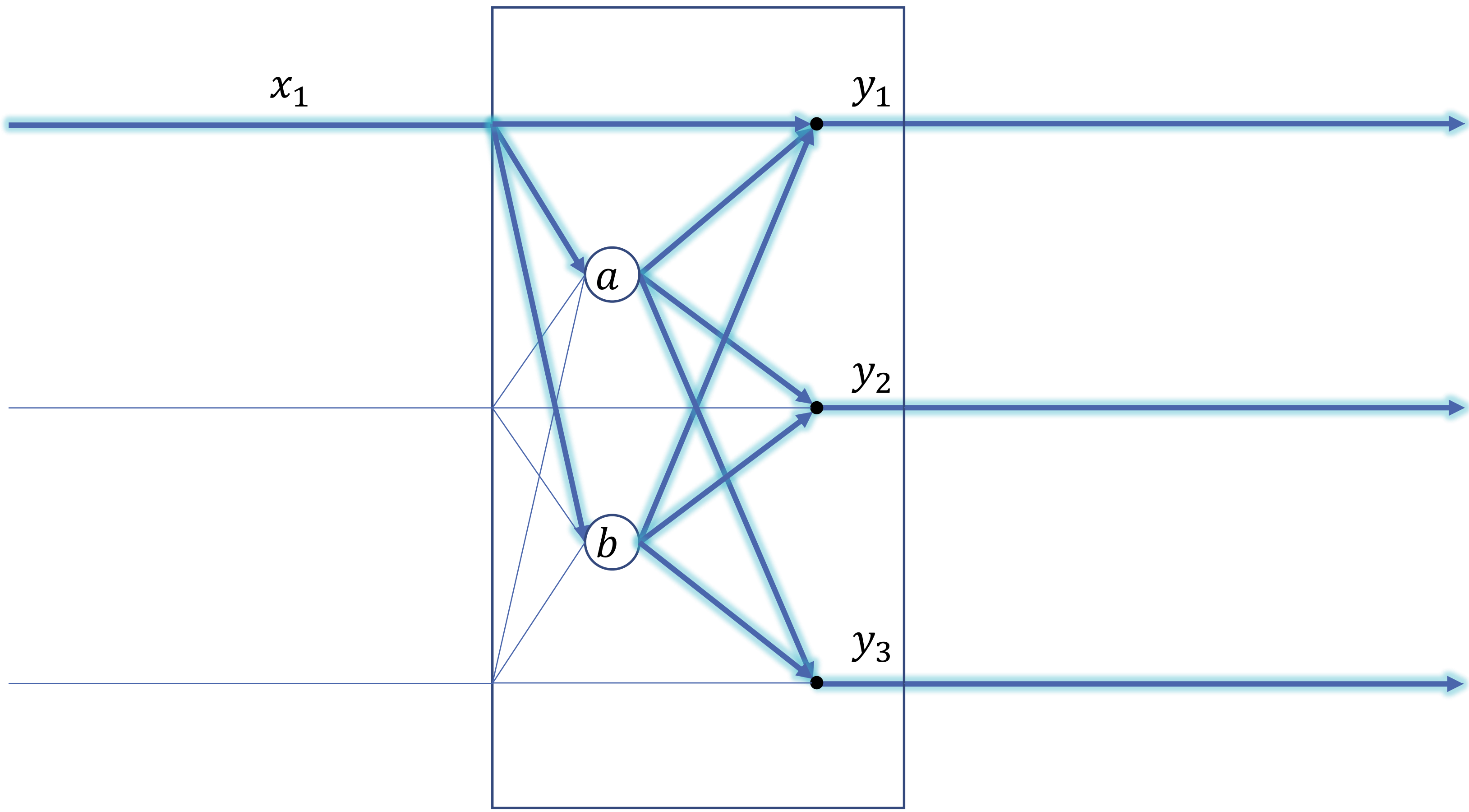
Notice x_1's different paths of influence

A New (Made Up) Activation in Town



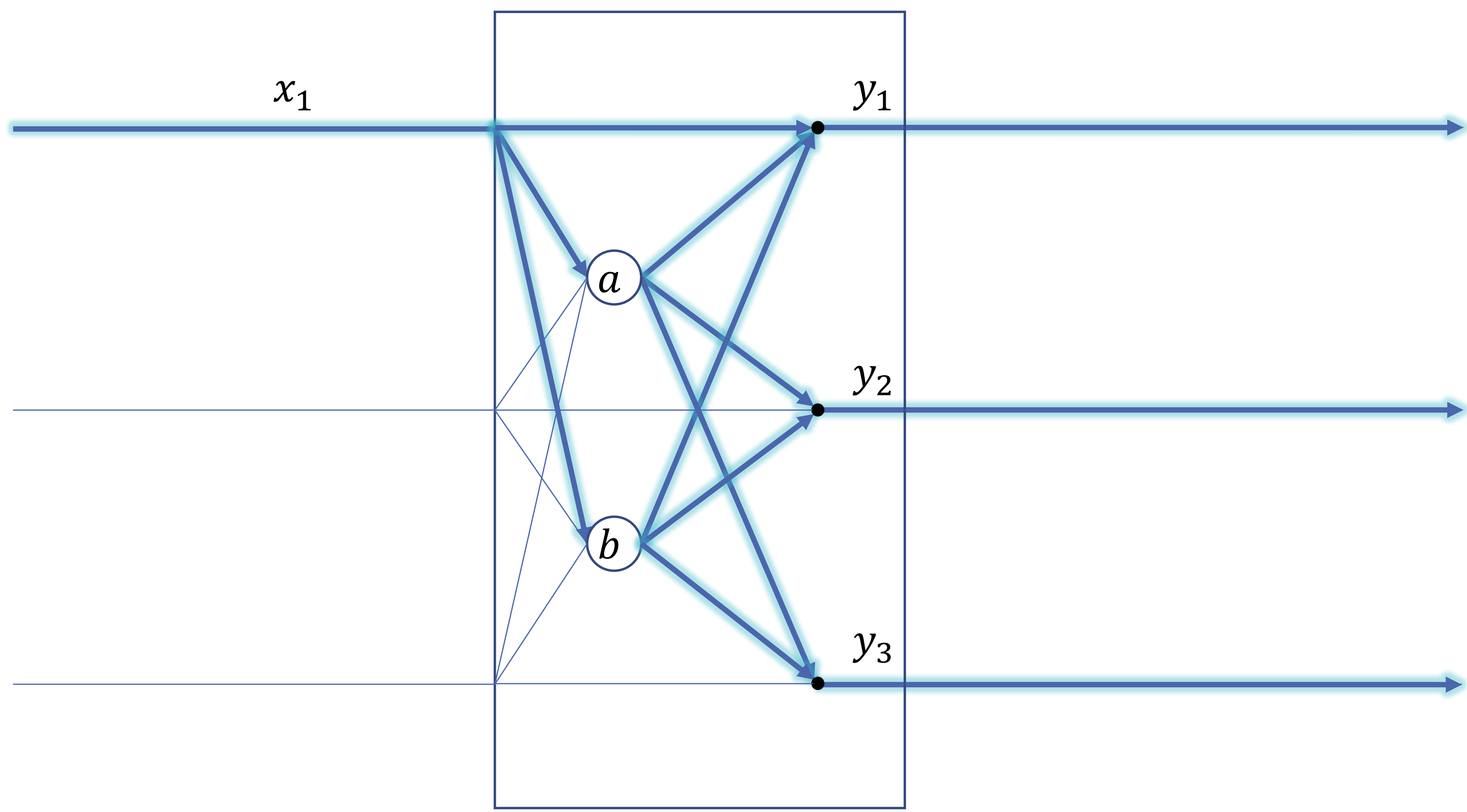
The derivative $\nabla_{x1}L$ is the sum of derivatives along these paths

A New (Made Up) Activation in Town



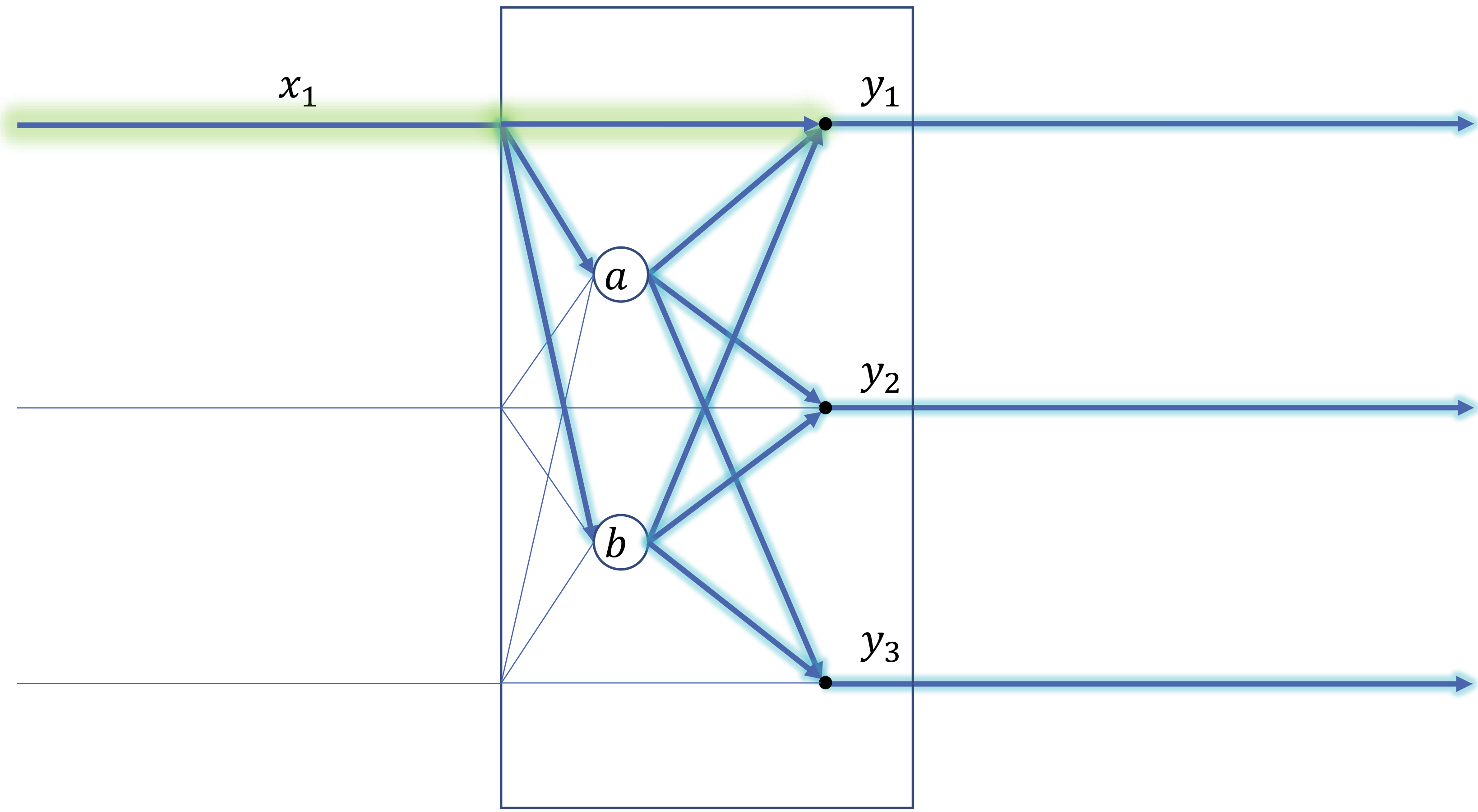
We will apply the chain rule at each node

A New (Made Up) Activation in Town



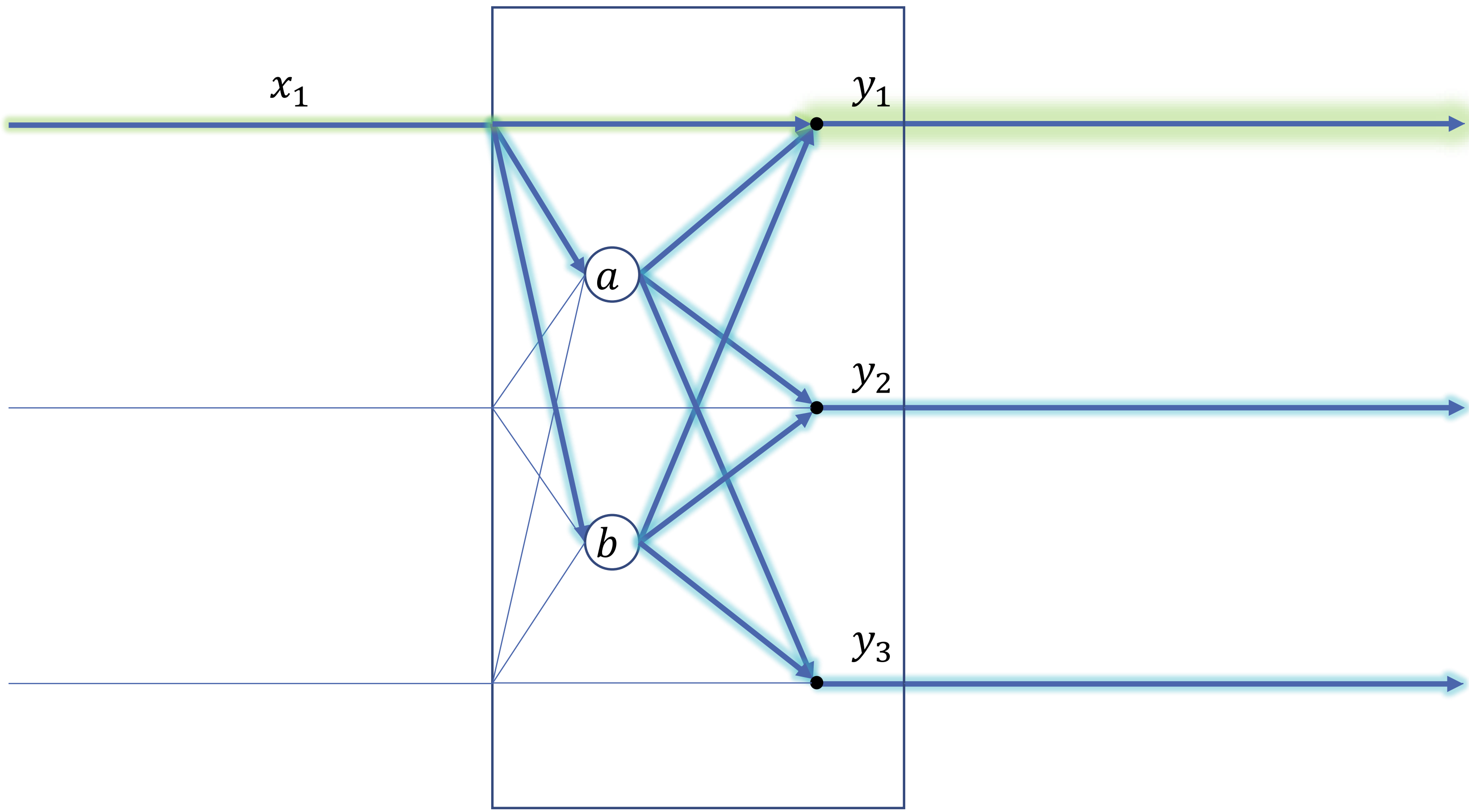
$\nabla_{x1} L =$

A New (Made Up) Activation in Town



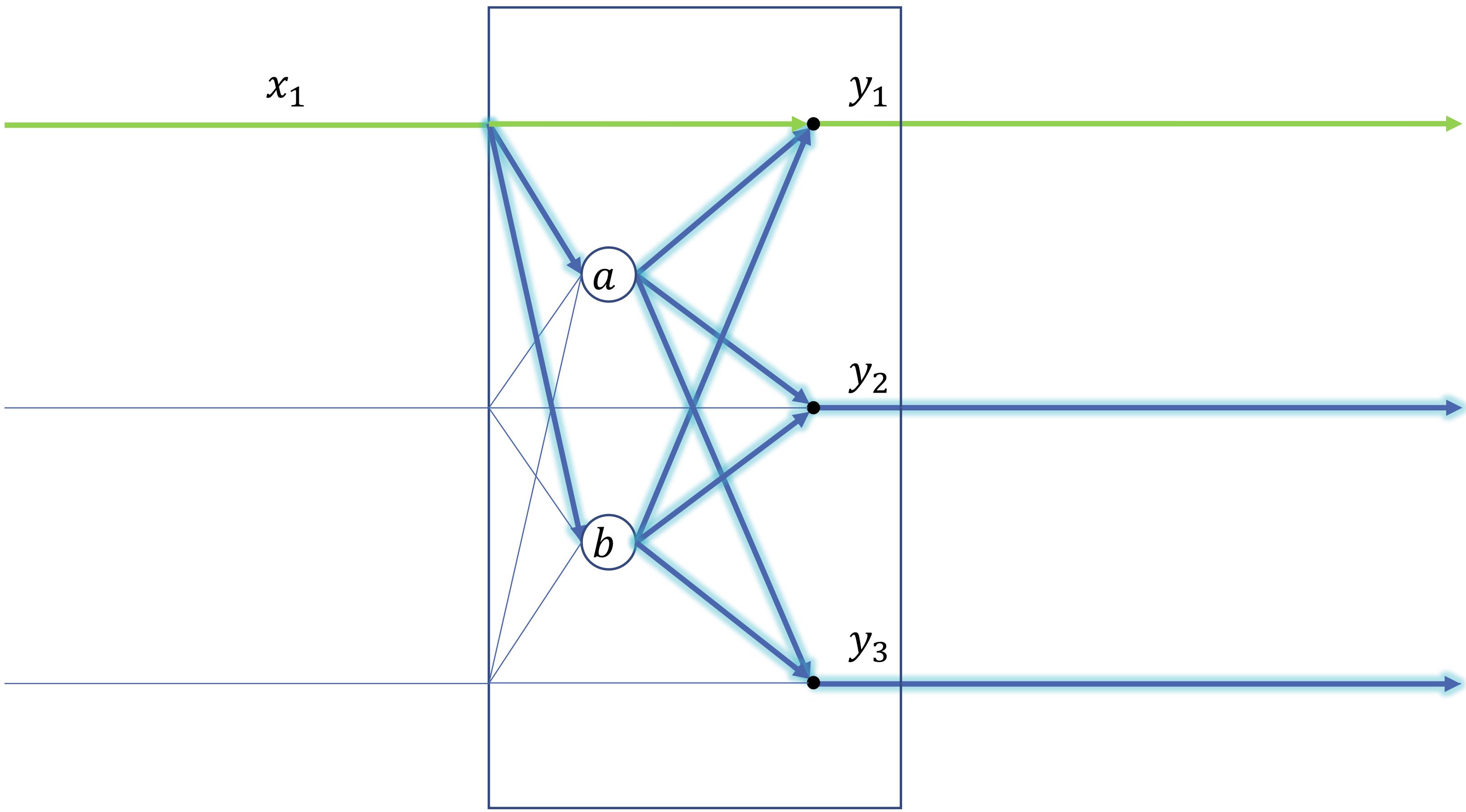
$$\nabla_{x1} L = \frac{dy_1}{dx_1}$$

A New (Made Up) Activation in Town



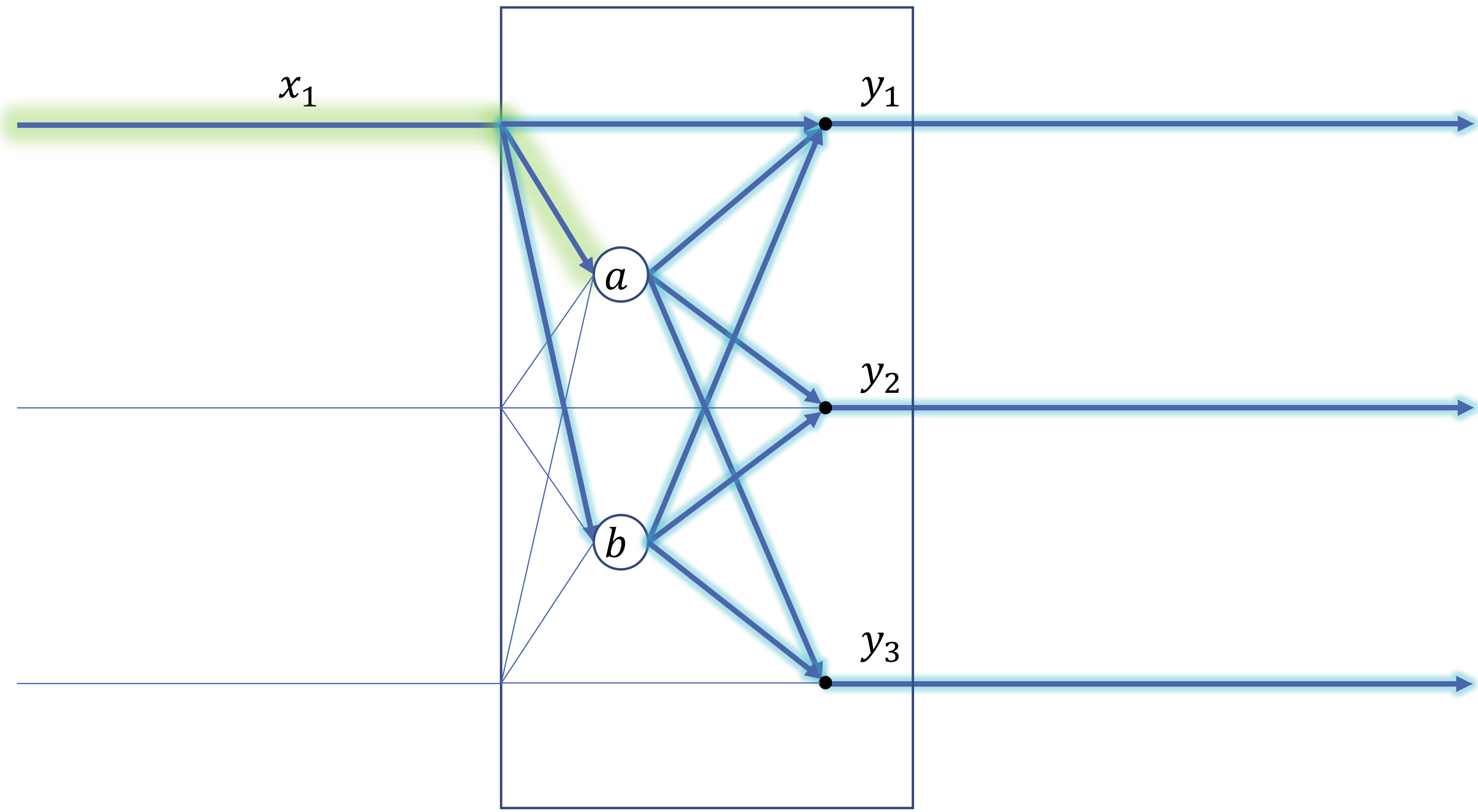
$$\nabla_{x1} L = \frac{dy_1}{dx_1} \frac{dL}{dy_1}$$

A New (Made Up) Activation in Town



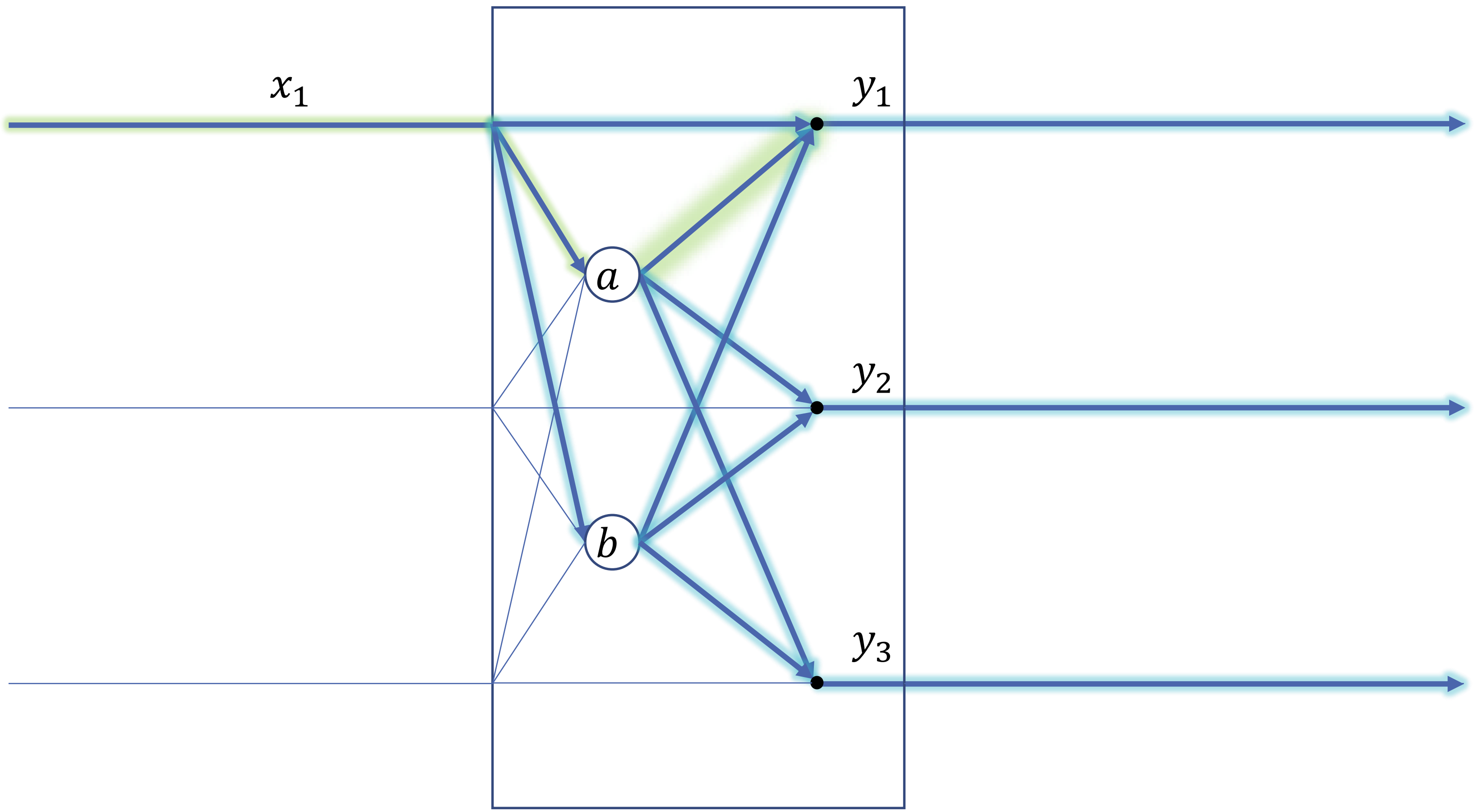
$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1}$$

A New (Made Up) Activation in Town



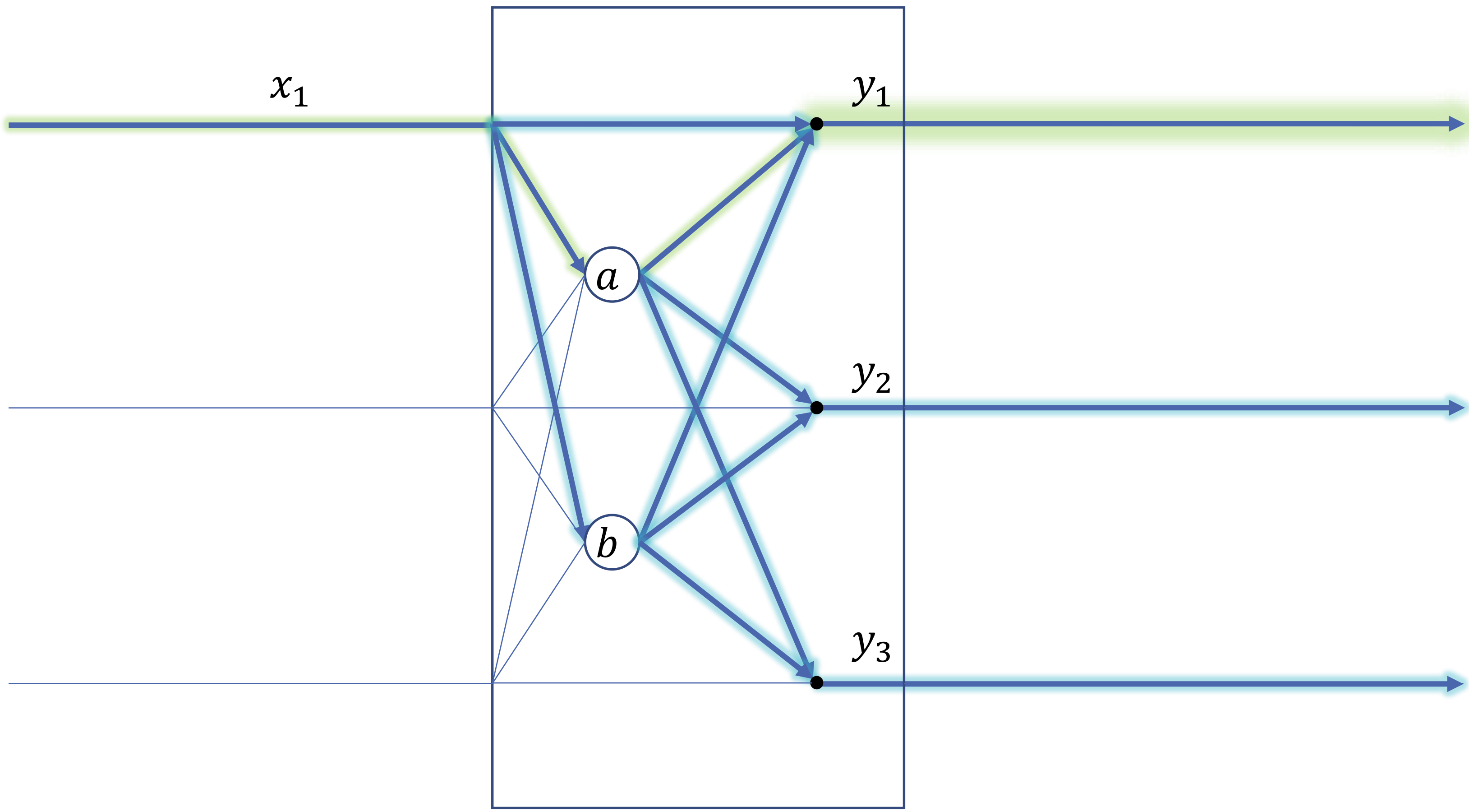
$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}$$

A New (Made Up) Activation in Town



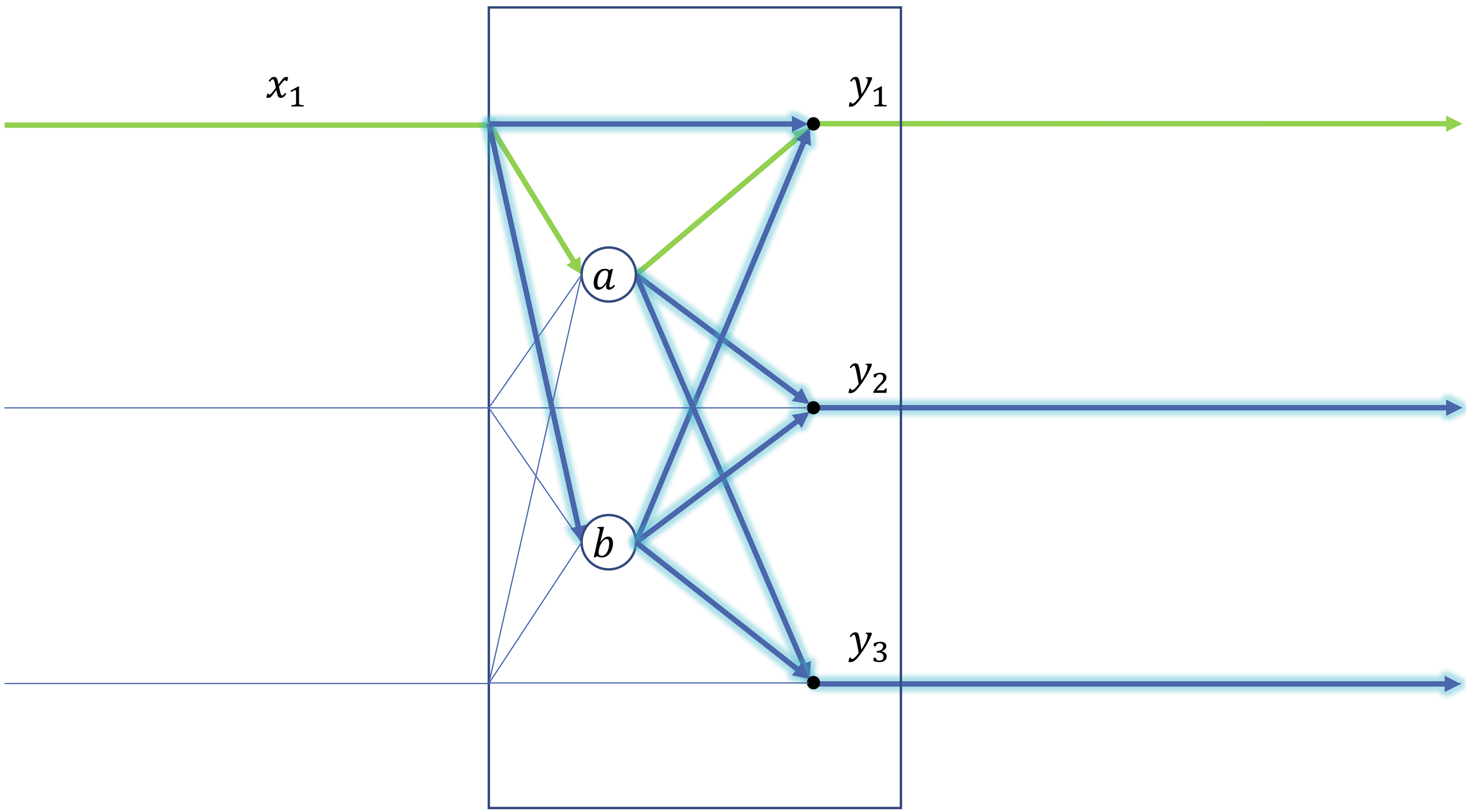
$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}$$

A New (Made Up) Activation in Town



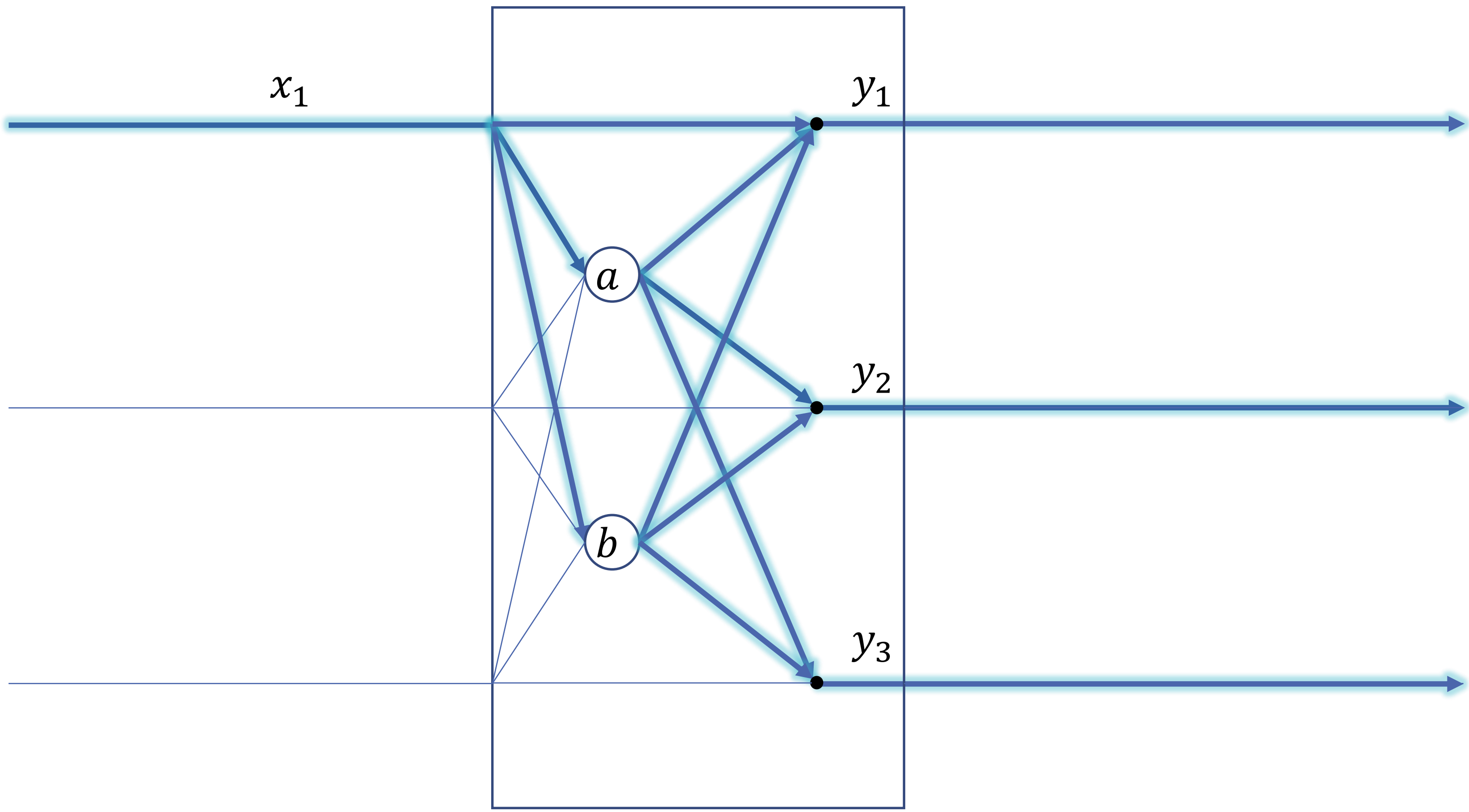
$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1}$$

A New (Made Up) Activation in Town



$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1}$$

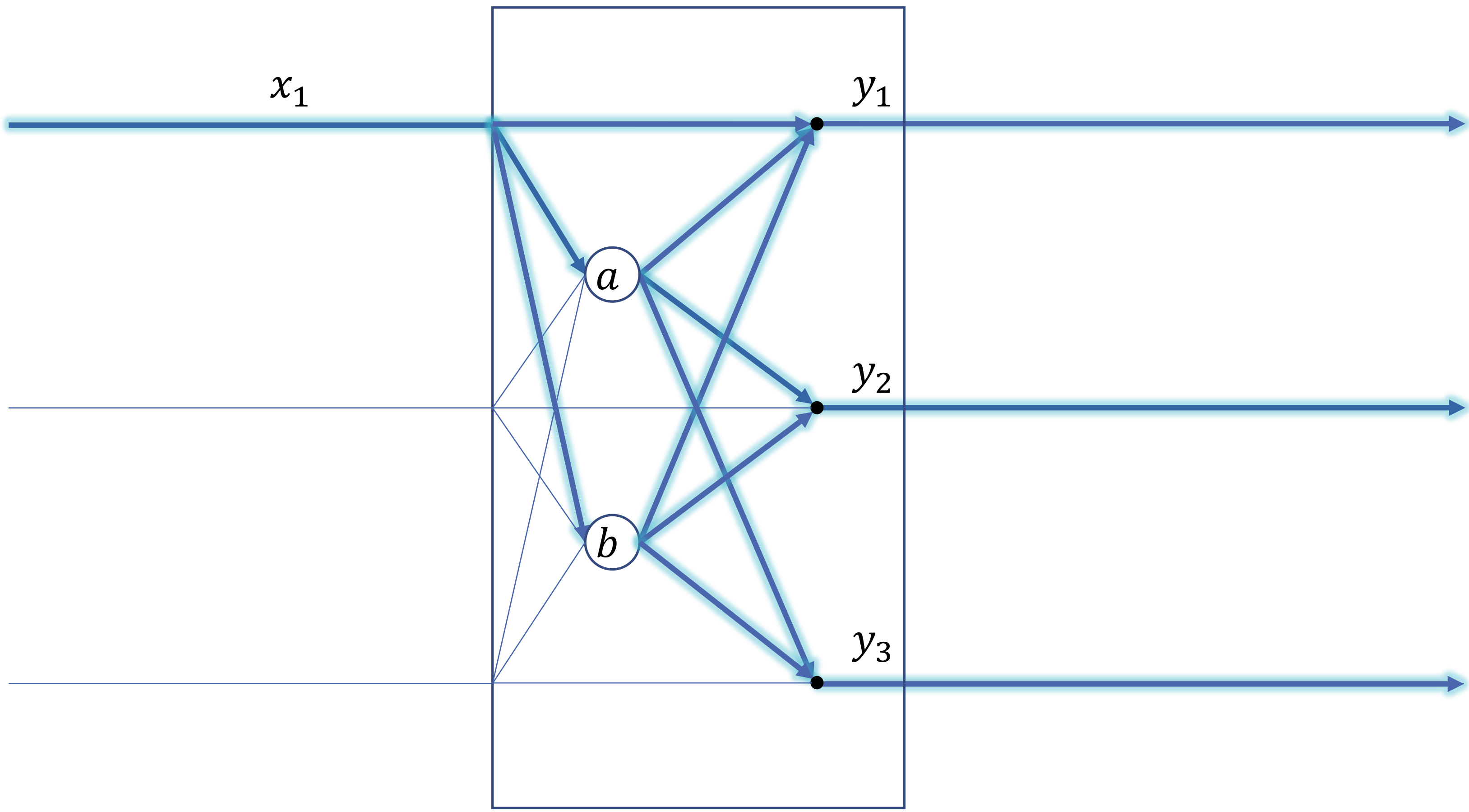
A New (Made Up) Activation in Town



$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \cdots$$

We can do this for the rest of the paths…

A New (Made Up) Activation in Town

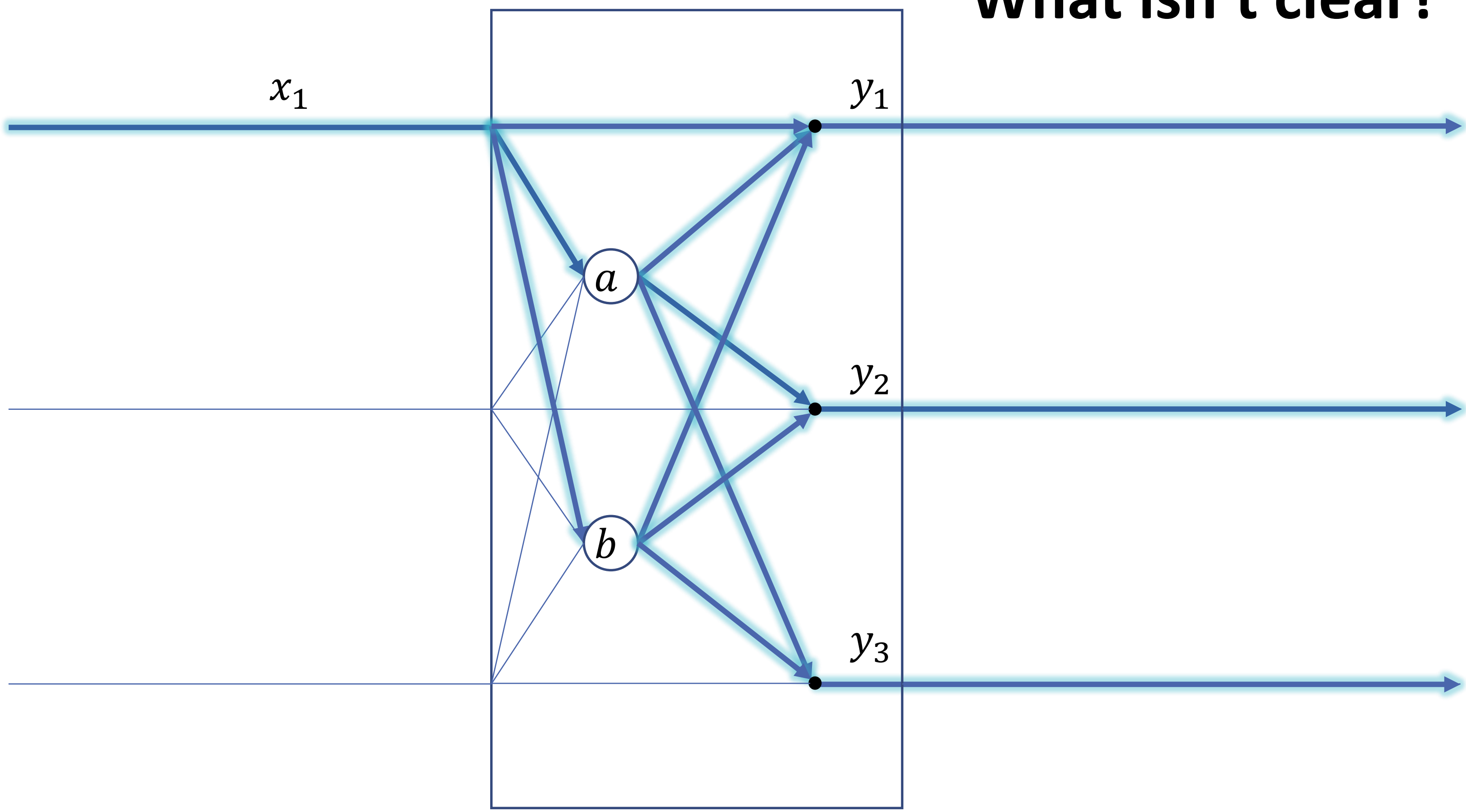


$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Seven terms, seven paths

A New (Made Up) Activation in Town

Questions?
What isn't clear?



$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Seven terms, seven paths

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Now we’re done with the influence diagram

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

$$a = \sum_j x_j \qquad b = \sum_j \ln(x_j) \qquad y_i = \cos\left(\frac{e^{x_i a}}{b}\right)$$

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

$$a = \sum_j x_j \qquad b = \sum_j \ln(x_j) \qquad y_i = \cos(\frac{e^{x_i\ a}}{b})$$

$$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1\ a}}{b})\frac{ae^{x_1\ a}}{b}$$

A New (Made Up) Activation in Town

$$\nabla_{x1}L = \frac{dy_1}{dx_1}\frac{\boldsymbol{dL}}{\boldsymbol{dy_1}} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

$$a = \sum_j x_j \qquad b = \sum_j \ln(x_j) \qquad y_i = \cos(\frac{e^{x_i\ a}}{b})$$

$$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1\ a}}{b})\frac{ae^{x_1\ a}}{b}$$

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

$a = \sum_j x_j \qquad b = \sum_j \ln(x_j) \qquad y_i = \cos(\frac{e^{x_i\ a}}{b})$

$$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1\ a}}{b})\frac{ae^{x_1\ a}}{b}$$

$$\frac{da}{dx_1} = 1$$

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{\boldsymbol{dy_1}}{\boldsymbol{da}}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

$a = \sum_j x_j$ $\qquad b = \sum_j \ln(x_j)$ $\qquad y_i = \cos(\frac{e^{x_i\ a}}{b})$

$$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1\ a}}{b})\frac{ae^{x_1\ a}}{b}$$

$$\frac{da}{dx_1} = 1$$

$$\frac{dy_1}{da} = -\sin(\frac{e^{x_1\ a}}{b})\frac{x_1 e^{x_1\ a}}{b}$$

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

$$a = \sum_j x_j \qquad b = \sum_j \ln(x_j) \qquad y_i = \cos\left(\frac{e^{x_i\ a}}{b}\right)$$

$$\frac{dy_1}{dx_1} = -\sin\left(\frac{e^{x_1\ a}}{b}\right)\frac{ae^{x_1\ a}}{b}$$

$$\frac{da}{dx_1} = 1$$

$$\frac{dy_1}{da} = -\sin\left(\frac{e^{x_1\ a}}{b}\right)\frac{x_1 e^{x_1\ a}}{b}$$

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

$$a = \sum_j x_j \qquad b = \sum_j \ln(x_j) \qquad y_i = \cos\left(\frac{e^{x_i\ a}}{b}\right)$$

$$\frac{dy_1}{dx_1} = -\sin\left(\frac{e^{x_1\ a}}{b}\right)\frac{ae^{x_1\ a}}{b}$$

$$\frac{da}{dx_1} = 1$$

$$\frac{dy_1}{da} = -\sin\left(\frac{e^{x_1\ a}}{b}\right)\frac{x_1 e^{x_1\ a}}{b}$$

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Time to calculate the necessary derivatives…

$$a = \sum_j x_j \qquad b = \sum_j \ln(x_j) \qquad y_i = \cos(\frac{e^{x_i\ a}}{b})$$

$$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1\ a}}{b})\frac{ae^{x_1\ a}}{b}$$

$$\frac{da}{dx_1} = 1$$

$$\frac{dy_1}{da} = -\sin(\frac{e^{x_1\ a}}{b})\frac{x_1 e^{x_1\ a}}{b}$$

$$\frac{db}{dx_1} = \frac{1}{x_1}$$

$$\frac{dy_1}{db} = \sin(\frac{e^{x_1\ a}}{b})\frac{x_1 e^{x_1\ a}}{b^2}$$

$$\frac{dy_2}{da} = -\sin(\frac{e^{x_2\ a}}{b})\frac{x_2 e^{x_2\ a}}{b}$$

$$\frac{dy_2}{db} = \sin(\frac{e^{x_2\ a}}{b})\frac{x_2 e^{x_2\ a}}{b^2}$$

$$\frac{dy_3}{da} = -\sin(\frac{e^{x_3\ a}}{b})\frac{x_3 e^{x_3\ a}}{b}$$

$$\frac{dy_3}{db} = \sin(\frac{e^{x_3\ a}}{b})\frac{x_3 e^{x_3\ a}}{b^2}$$

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

$$\frac{dy_1}{dx_1} = -\sin\left(\frac{e^{x_1 a}}{b}\right)\frac{a e^{x_1 a}}{b}$$

$$\frac{da}{dx_1} = 1$$

$$\frac{dy_1}{da} = -\sin\left(\frac{e^{x_1 a}}{b}\right)\frac{x_1 e^{x_1 a}}{b}$$

$$\frac{db}{dx_1} = \frac{1}{x_1}$$

$$\frac{dy_1}{db} = \sin\left(\frac{e^{x_1 a}}{b}\right)\frac{x_1 e^{x_1 a}}{b^2}$$

$$\frac{dy_2}{da} = -\sin\left(\frac{e^{x_2 a}}{b}\right)\frac{x_2 e^{x_2 a}}{b}$$

$$\frac{dy_2}{db} = \sin\left(\frac{e^{x_2 a}}{b}\right)\frac{x_2 e^{x_2 a}}{b^2}$$

$$\frac{dy_3}{da} = -\sin\left(\frac{e^{x_3 a}}{b}\right)\frac{x_3 e^{x_3 a}}{b}$$

$$\frac{dy_3}{db} = \sin\left(\frac{e^{x_3 a}}{b}\right)\frac{x_3 e^{x_3 a}}{b^2}$$

Now we plug things in / simplify

A New (Made Up) Activation in Town

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

$$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1\,a}}{b})\frac{ae^{x_1\,a}}{b}$$

$$\frac{da}{dx_1} = 1$$

$$\frac{dy_1}{da} = -\sin(\frac{e^{x_1\,a}}{b})\frac{x_1e^{x_1\,a}}{b}$$

$$\frac{db}{dx_1} = \frac{1}{x_1}$$

$$\frac{dy_1}{db} = \sin(\frac{e^{x_1\,a}}{b})\frac{x_1e^{x_1\,a}}{b^2}$$

$$\frac{dy_2}{da} = -\sin(\frac{e^{x_2\,a}}{b})\frac{x_2e^{x_2\,a}}{b}$$

$$\frac{dy_2}{db} = \sin(\frac{e^{x_2\,a}}{b})\frac{x_2e^{x_2\,a}}{b^2}$$

$$\frac{dy_3}{da} = -\sin(\frac{e^{x_3\,a}}{b})\frac{x_3e^{x_3\,a}}{b}$$

$$\frac{dy_3}{db} = \sin(\frac{e^{x_3\,a}}{b})\frac{x_3e^{x_3\,a}}{b^2}$$

Now we plug things in / simplify

“The simplification is left as an exercise to the reader”

Influence Diagrams

A little painful, but algorithmic:

Break things up

$$a = \sum_j x_j \quad b = \sum_j \ln(x_j) \quad y_i = \cos(\frac{e^{x_i\ a}}{b})$$

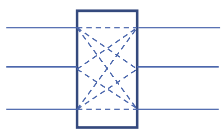
Influence Diagrams

A little painful, but algorithmic:

Break things up

$$a = \sum_j x_j \quad b = \sum_j \ln(x_j) \quad y_i = \cos(\frac{e^{x_i \ a}}{b})$$

Draw the influence diagram



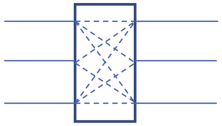
Influence Diagrams

A little painful, but algorithmic:

Break things up

$$a = \sum_j x_j \quad b = \sum_j \ln(x_j) \quad y_i = \cos\left(\frac{e^{x_i}\ a}{b}\right)$$

Draw the influence diagram



Write out paths using the diagram / chain rule

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

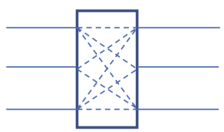
Influence Diagrams

A little painful, but algorithmic:

Break things up

$$a = \sum_j x_j \quad b = \sum_j \ln(x_j) \quad y_i = \cos(\frac{e^{x_i\ a}}{b})$$

Draw the influence diagram



Write out paths using the diagram / chain rule

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Calculate necessary derivatives

$$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1\ a}}{b})\frac{ae^{x_1\ a}}{b} \quad \frac{da}{dx_1} = 1 \quad \frac{dy_1}{da} = -\sin(\frac{e^{x_1\ a}}{b})\frac{x_1 e^{x_1\ a}}{b} \quad \frac{db}{dx_1} = \frac{1}{x_1} \quad \frac{dy_1}{db} = \sin(\frac{e^{x_1\ a}}{b})\frac{x_1 e^{x_1\ a}}{b^2}$$ etc...

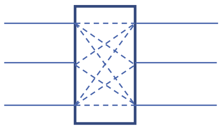
Influence Diagrams

A little painful, but algorithmic:

Break things up

$$a = \sum_j x_j \quad b = \sum_j \ln(x_j) \quad y_i = \cos(\frac{e^{x_i\ a}}{b})$$

Draw the influence diagram



Write out paths using the diagram / chain rule

$$\nabla_{x1}L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Calculate necessary derivatives

$$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1\ a}}{b})\frac{ae^{x_1\ a}}{b} \quad \frac{da}{dx_1} = 1 \quad \frac{dy_1}{da} = -\sin(\frac{e^{x_1\ a}}{b})\frac{x_1e^{x_1\ a}}{b} \quad \frac{db}{dx_1} = \frac{1}{x_1} \quad \frac{dy_1}{db} = \sin(\frac{e^{x_1\ a}}{b})\frac{x_1e^{x_1\ a}}{b^2}$$ etc...

Plug things in / simplify

A mess 😁

Influence Diagrams

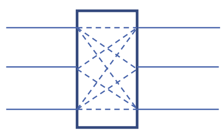
Questions?
What isn't clear?

A little painful, but algorithmic:

Break things up

$$a = \sum_j x_j \quad b = \sum_j \ln(x_j) \quad y_i = \cos(\frac{e^{x_i} a}{b})$$

Draw the influence diagram



Write out paths using the diagram / chain rule

$$\nabla_{x1} L = \frac{dy_1}{dx_1}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_1}{da}\frac{dL}{dy_1} + \frac{db}{dx_1}\frac{dy_1}{db}\frac{dL}{dy_1} + \frac{da}{dx_1}\frac{dy_2}{da}\frac{dL}{dy_2} + \frac{db}{dx_1}\frac{dy_2}{db}\frac{dL}{dy_2} + \frac{da}{dx_1}\frac{dy_3}{da}\frac{dL}{dy_3} + \frac{db}{dx_1}\frac{dy_3}{db}\frac{dL}{dy_3}$$

Calculate necessary derivatives

$\frac{dy_1}{dx_1} = -\sin(\frac{e^{x_1} a}{b})\frac{ae^{x_1} a}{b}$ $\quad \frac{da}{dx_1} = 1$ $\quad \frac{dy_1}{da} = -\sin(\frac{e^{x_1} a}{b})\frac{x_1 e^{x_1} a}{b}$ $\quad \frac{db}{dx_1} = \frac{1}{x_1}$ $\quad \frac{dy_1}{db} = \sin(\frac{e^{x_1} a}{b})\frac{x_1 e^{x_1} a}{b^2}$ etc…

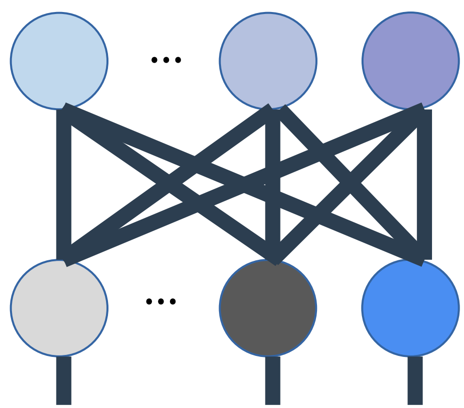
Plug things in / simplify

A mess 😁

Computational Graphs

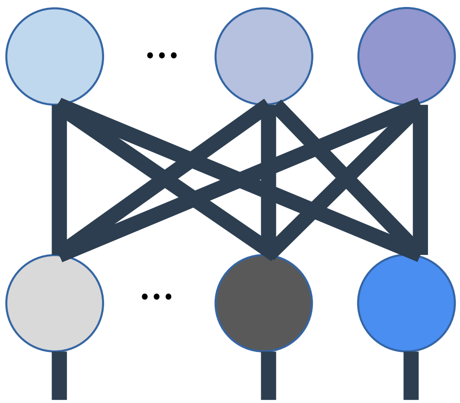
Feel free to follow the backprop part on paper.

Simple MLP



An MLP with one tanh activated hidden layer

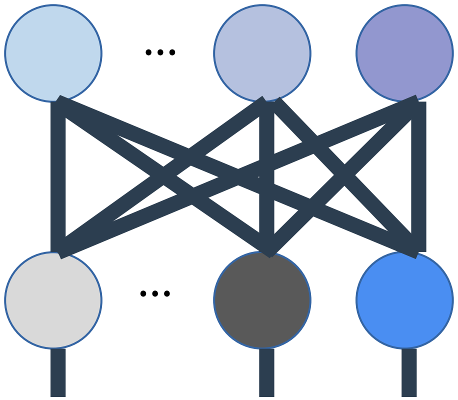
Simple MLP



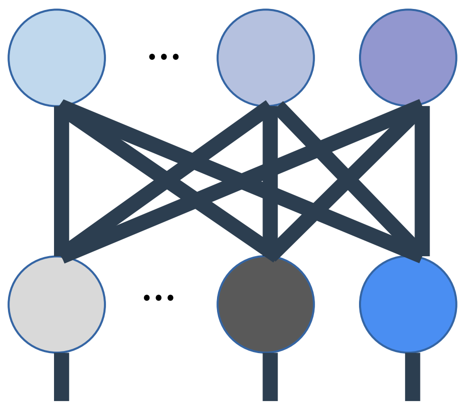
We want to easily compute the derivative
with respect to the weights Wi and biases bi

Simple MLP

Linear	$z = W_1 x + b_1$
Activation	$\mathrm{out} = \tanh(z)$



Simple MLP

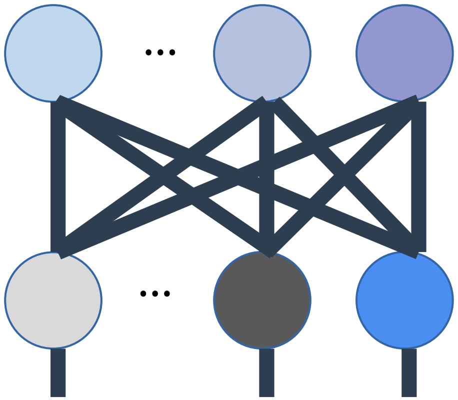


$$z = W_1 x + b_1$$
$$\text{out} = \tanh(z)$$

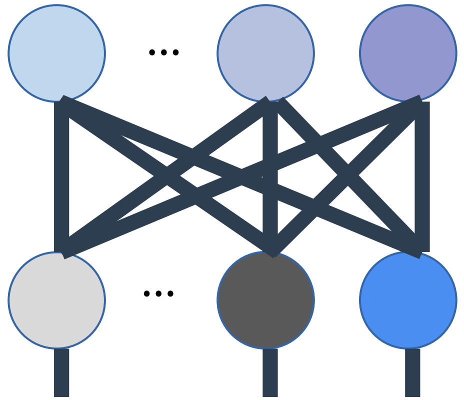
Let's unravel these equations
into **unary** and **binary** operations
(one or two arguments only)

Simple MLP

Linear	$z_1 = W_1 x$ $z_2 = z_1 + b_1$
Activation	$\text{out} = \tanh(z_2)$



Simple MLP



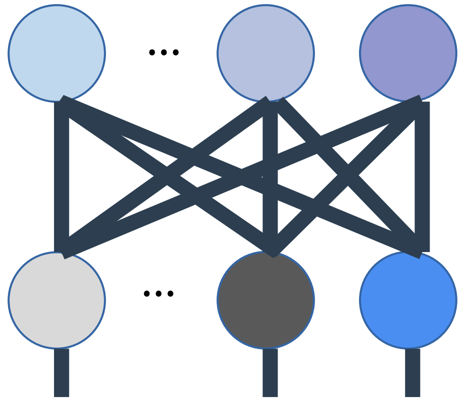
$$z_1 = W_1 x$$
$$z_2 = z_1 + b_1$$
$$\text{out} = \tanh(z_2)$$

This allows us to **reuse rules for propagating derivatives** through simple functions like +, *

Simple MLP

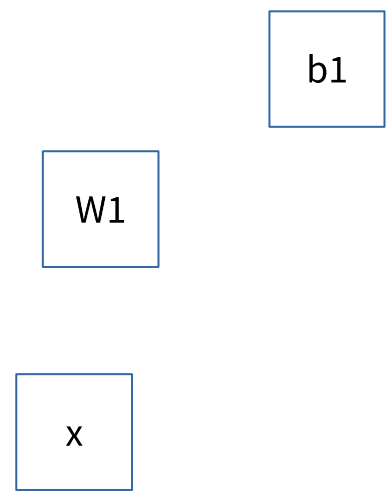
$$z_1 = W_1 x$$
$$z_2 = z_1 + b_1$$
$$\text{out} = \tanh(z_2)$$

Now let's step through this to create a **computational graph** (forward pass)



Simple MLP

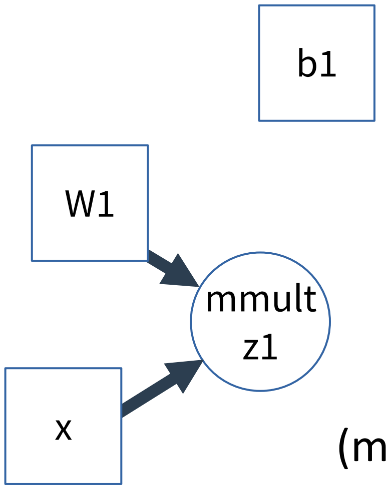
$$z_1 = W_1 x$$
$$z_2 = z_1 + b_1$$
$$\text{out} = \tanh(z_2)$$



Our initial variables

Simple MLP

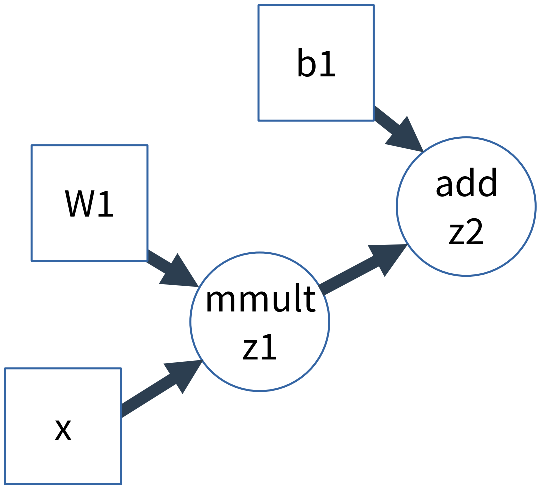
$$z_1 = W_1 x$$
$$z_2 = z_1 + b_1$$
$$\text{out} = \tanh(z_2)$$



(matrix multiply)

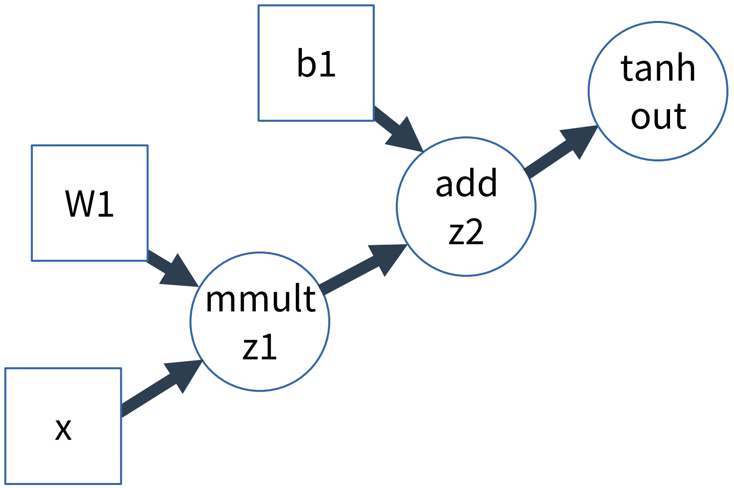
Simple MLP

$$z_1 = W_1 x$$
$$z_2 = z_1 + b_1$$
$$\text{out} = \tanh(z_2)$$



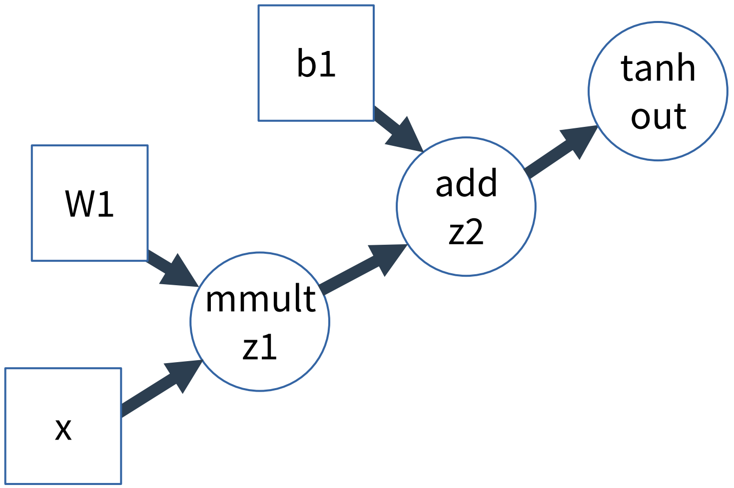
Simple MLP

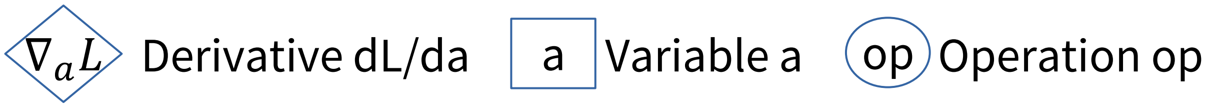
$$z_1 = W_1 x$$
$$z_2 = z_1 + b_1$$
$$\text{out} = \tanh(z_2)$$



Simple MLP

$$z_1 = W_1 x$$
$$z_2 = z_1 + b_1$$
$$\text{out} = \tanh(z_2)$$

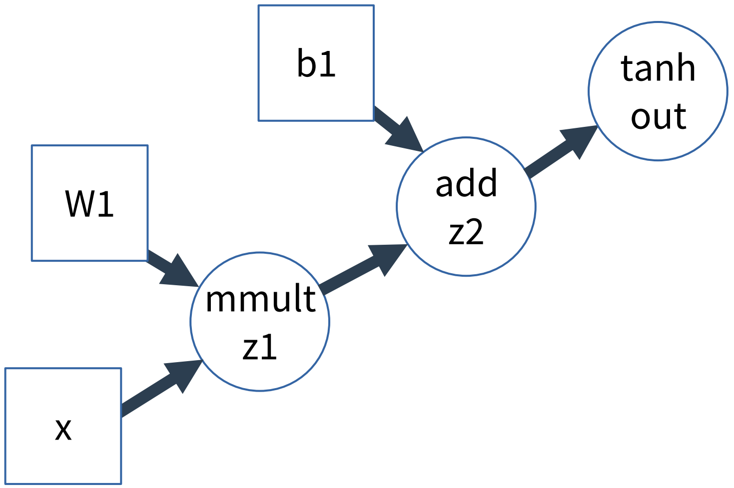


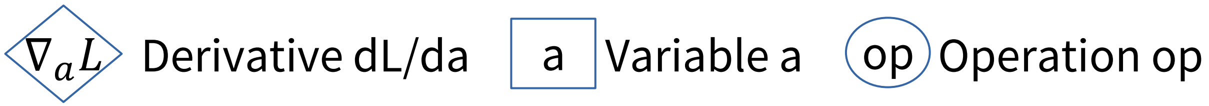


Simple MLP

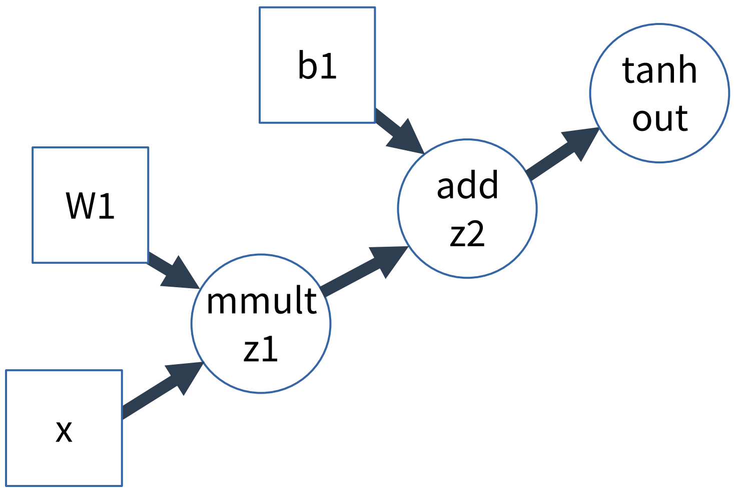
Questions?
What isn't clear?

$$z_1 = W_1 x$$
$$z_2 = z_1 + b_1$$
$$\mathrm{out} = \tanh(z_2)$$





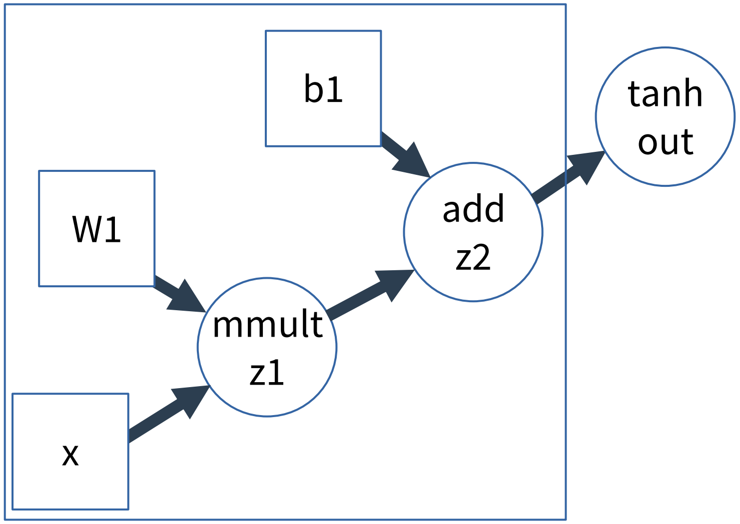
Simple MLP



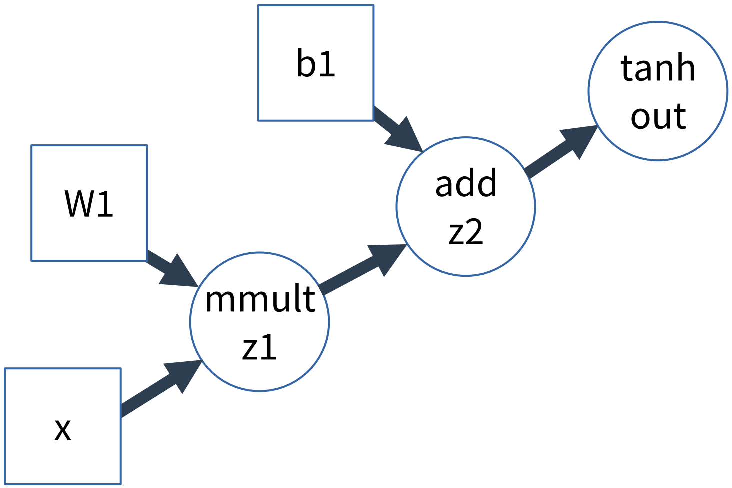
$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

Linear (just for reference)

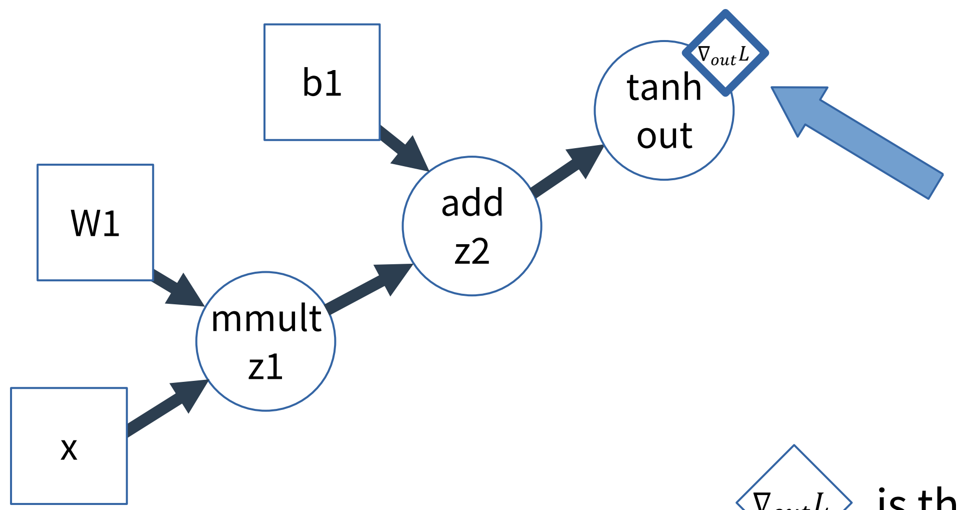


Simple MLP


b1
tanh
out
W1
add
z2
mmult
z1
x


$\nabla_a L$ Derivative dL/da
a Variable a
op Operation

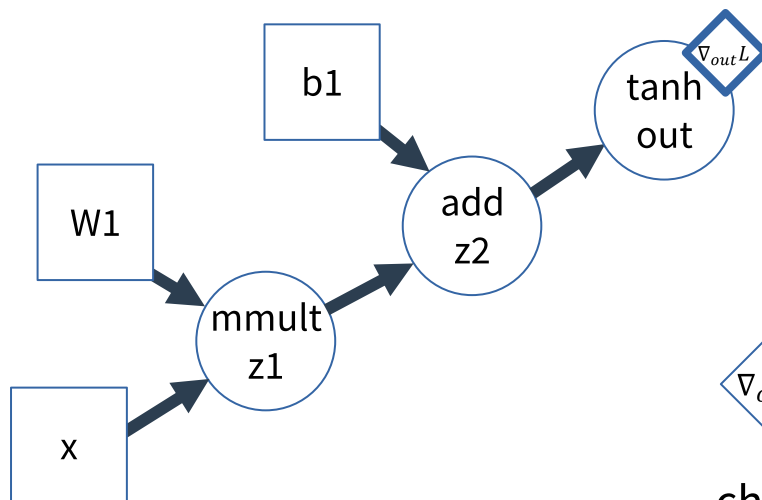
Simple MLP



$\nabla_{out}L$ is the derivative of the loss function with respect to the output.

$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

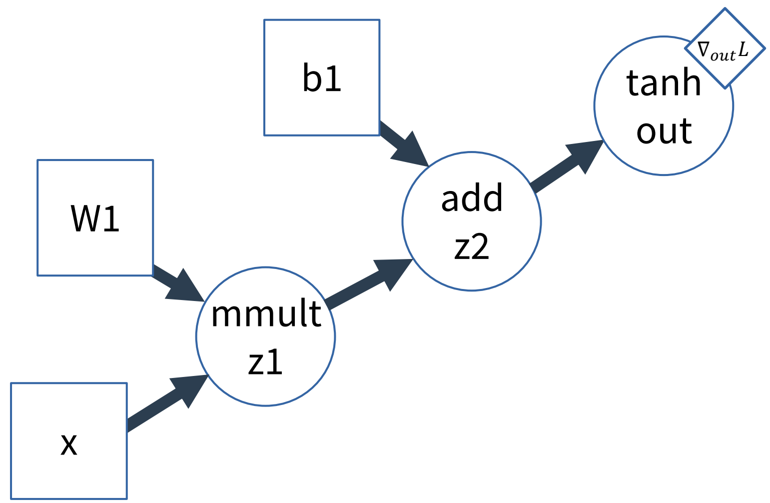


$\nabla_{out}L$ is calculated from our chosen loss function.

For simplicity, this example does not have the loss computation in the graph.

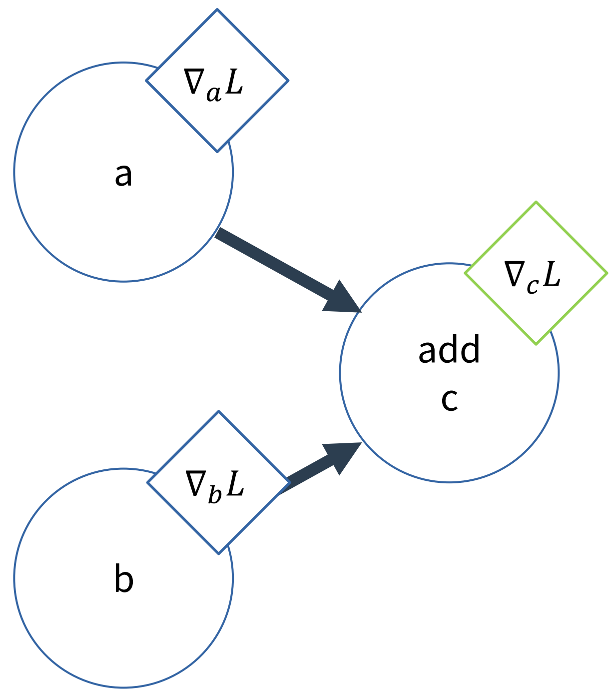
$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP



$\nabla_a L$ Derivative dL/da a Variable a op Operation

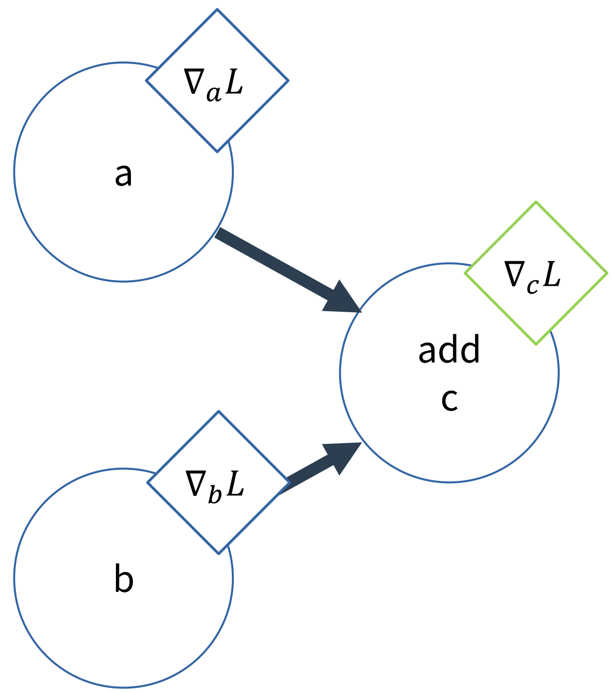
Aside: Backward Functions



Given $\nabla_c L$, what are $\nabla_a L$ and $\nabla_a L$ in terms of the chain rule?

$\nabla_a L$ Derivative dL/da a Variable a op Operation

Aside: Backward Functions

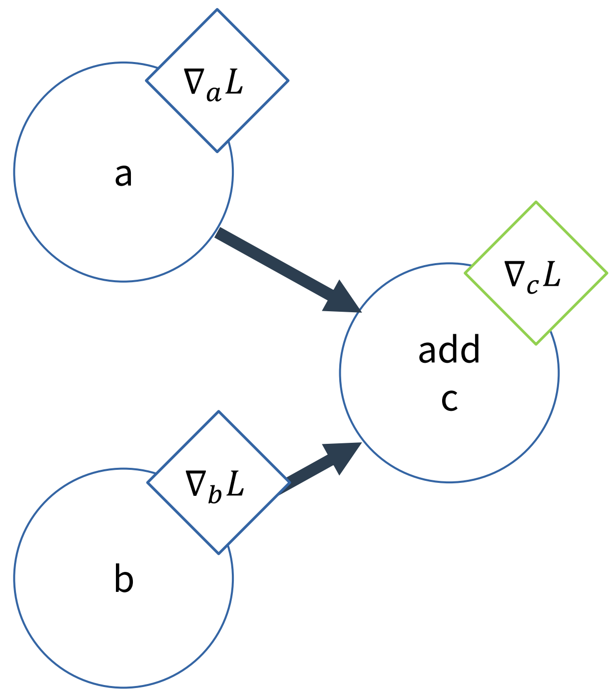


Given $\nabla_c L$, what are $\nabla_a L$ and $\nabla_a L$ in terms of the chain rule?

$$\frac{d\mathrm{L}}{d\mathrm{a}} = \frac{d\mathrm{L}}{d\mathrm{c}} * \frac{d\mathrm{c}}{d\mathrm{a}}$$

$\nabla_a L$ Derivative dL/da   a Variable a   op Operation

Aside: Backward Functions

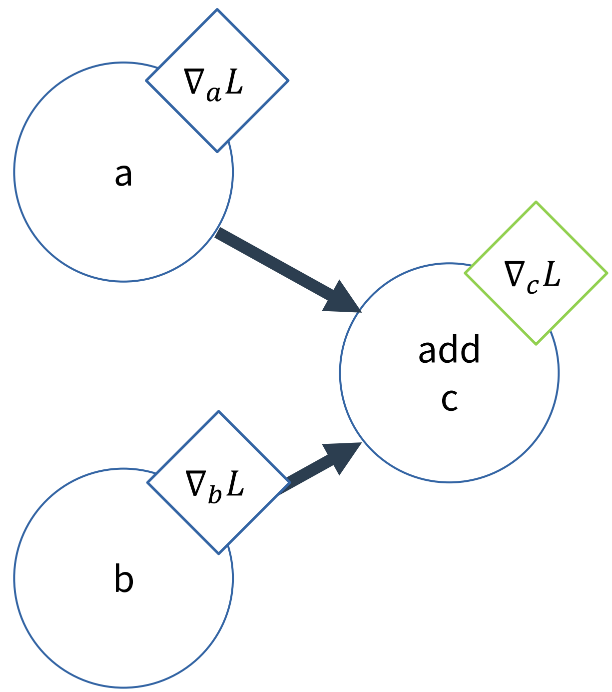


Given $\nabla_c L$, what are $\nabla_a L$ and $\nabla_a L$ in terms of the chain rule?

$$\frac{d\mathrm{L}}{d\mathrm{a}} = \frac{d\mathrm{L}}{d\mathrm{c}} * \frac{d\mathrm{c}}{d\mathrm{a}} \qquad \frac{d\mathrm{L}}{d\mathrm{b}} = \frac{d\mathrm{L}}{d\mathrm{c}} * \frac{d\mathrm{c}}{d\mathrm{b}}$$

$\nabla_a L$ Derivative dL/da a Variable a op Operation

Aside: Backward Functions

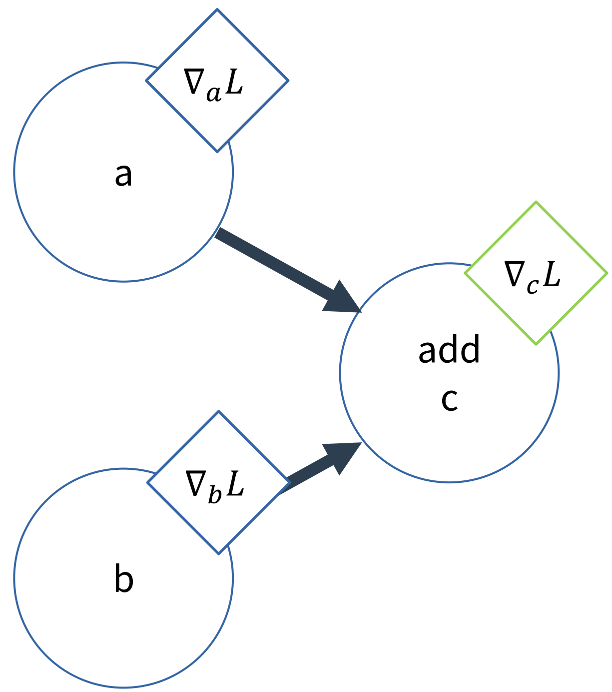


What do these simplify to? Hint: c=a+b, what's dc/da

$$\frac{d\mathrm{L}}{d\mathrm{a}} = \frac{d\mathrm{L}}{d\mathrm{c}} * \frac{d\mathrm{c}}{d\mathrm{a}} \qquad \frac{d\mathrm{L}}{d\mathrm{b}} = \frac{d\mathrm{L}}{d\mathrm{c}} * \frac{d\mathrm{c}}{d\mathrm{b}}$$

$\nabla_a L$ Derivative dL/da a Variable a op Operation

Aside: Backward Functions

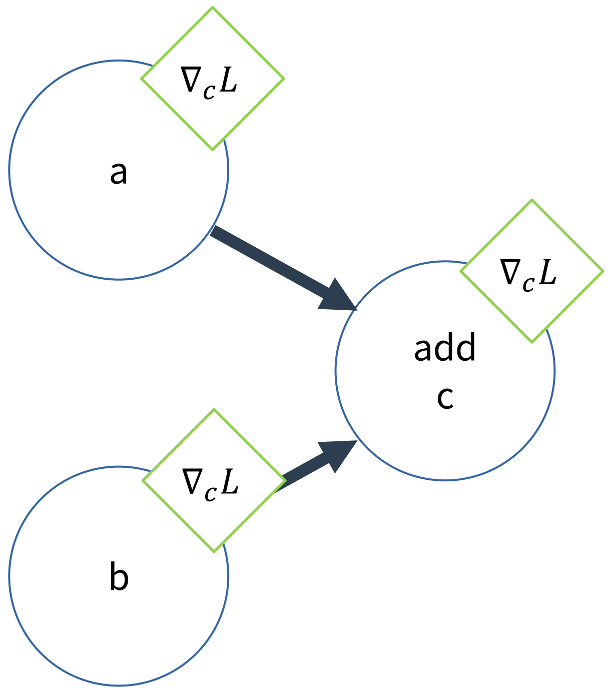


What do these simplify to? Hint: c=a+b, what's dc/da

$$\frac{dL}{da} = \frac{dL}{dc} * \cancel{\frac{dc}{da}} = \frac{dL}{dc} \qquad \frac{dL}{db} = \frac{dL}{dc} * \cancel{\frac{dc}{db}} = \frac{dL}{dc}$$

$\nabla_a L$ Derivative dL/da a Variable a op Operation

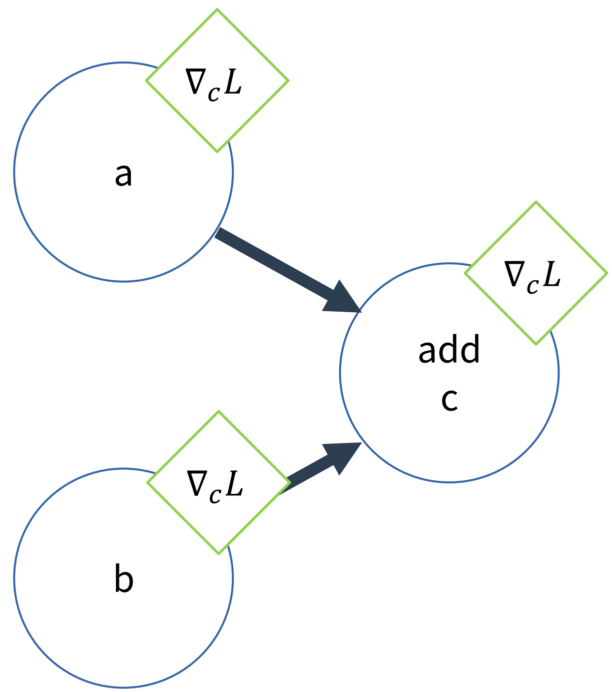
Aside: Backward Functions



Add’s **backward function** is to pass the gradient back unchanged

Aside: Backward Functions

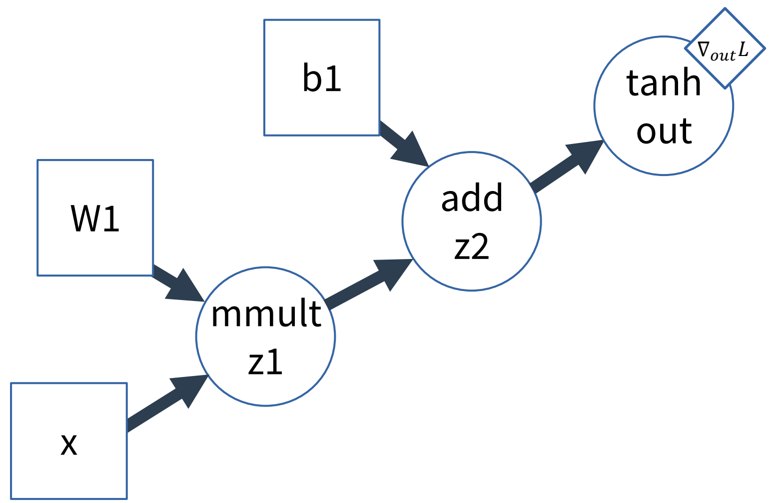
Questions?
What isn't clear?



Add's **backward function** is to pass the gradient back unchanged

$\nabla_a L$ Derivative dL/da | a Variable a | op Operation

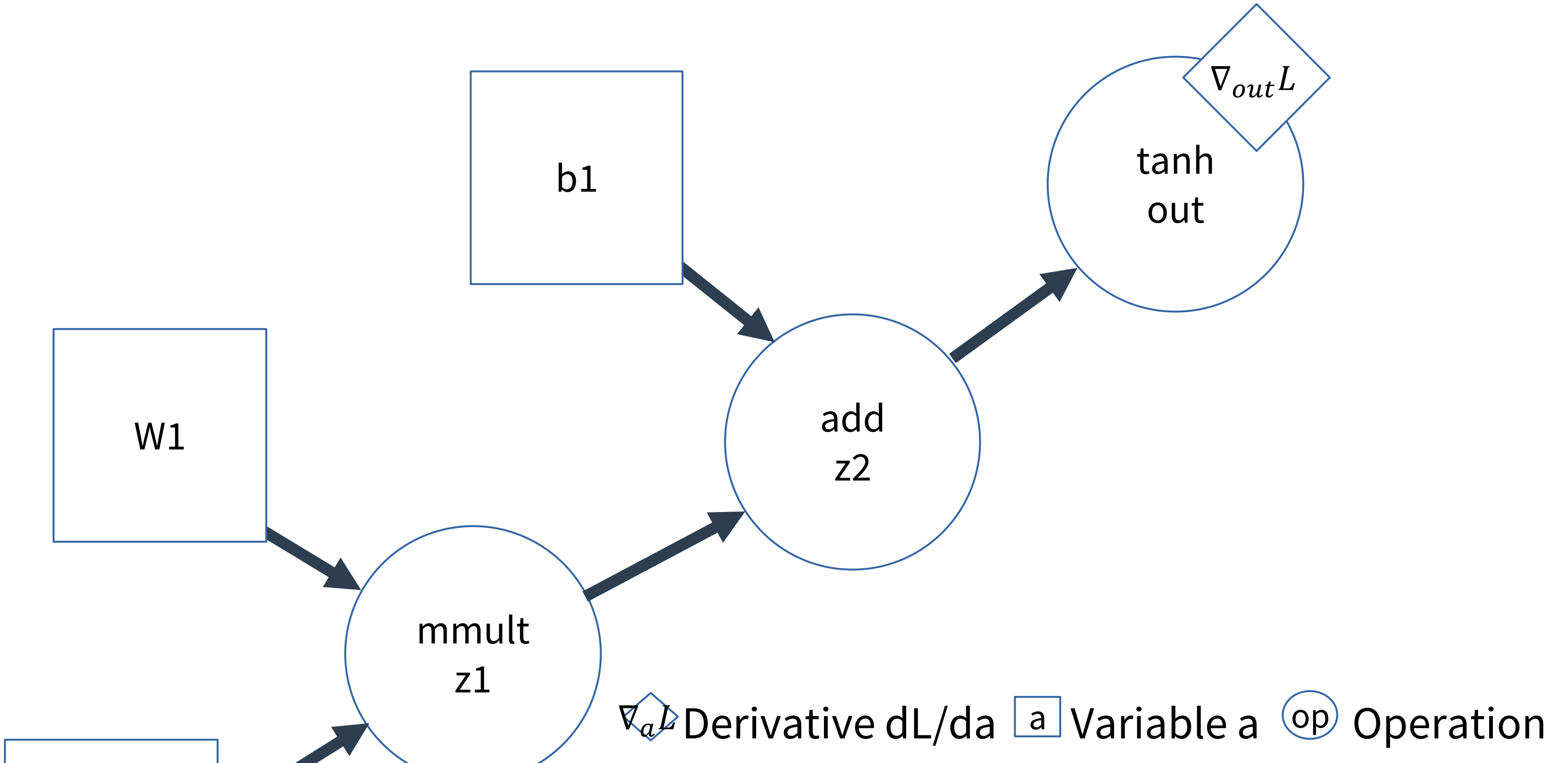
Simple MLP


b1
tanh
out
$\nabla_{out} L$
W1
add
z2
mmult
z1
x


$\nabla_a L$ Derivative dL/da
a Variable a
op Operation

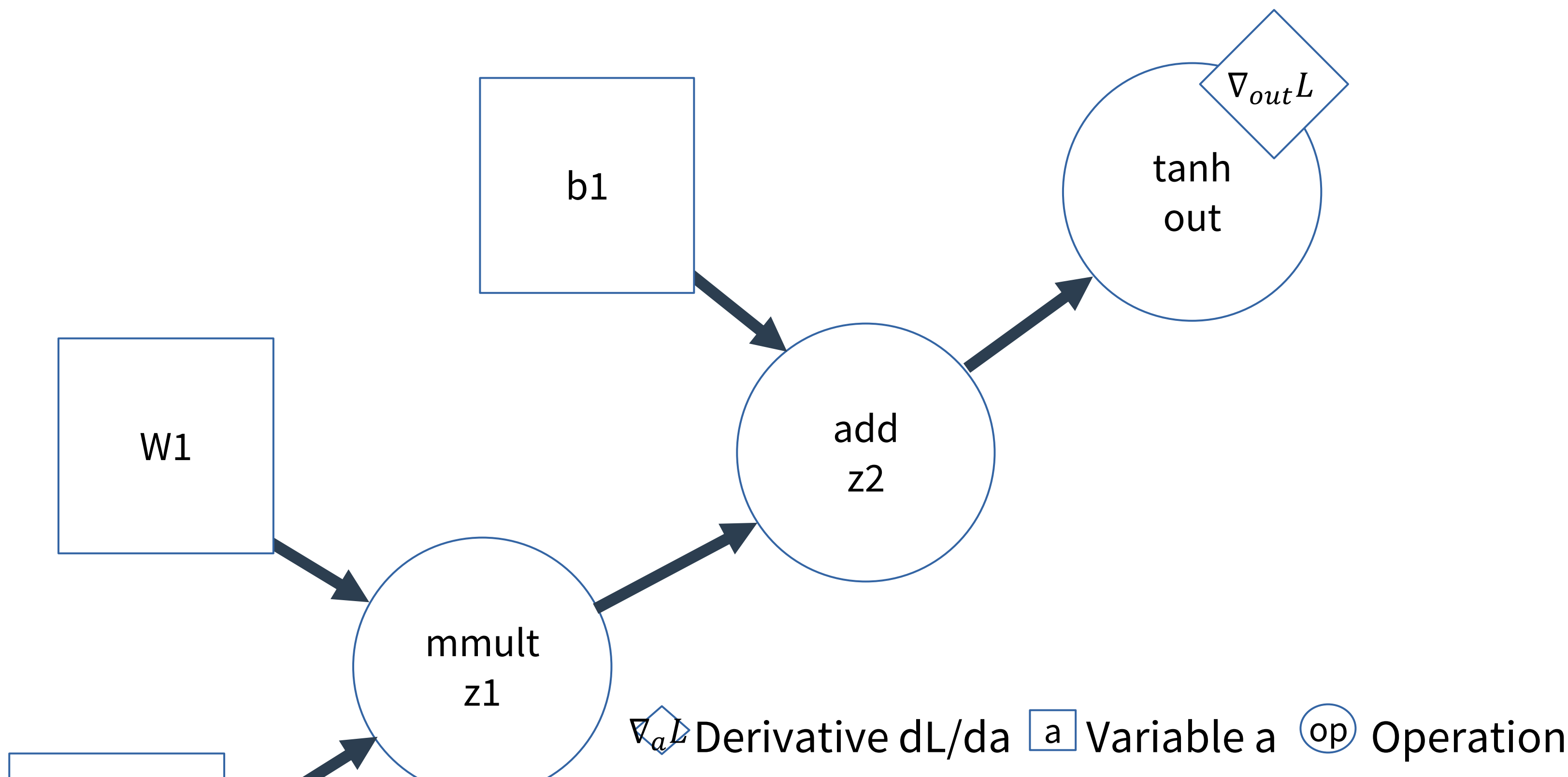
Simple MLP

We will perform a graph search from the end, updating derivatives as we go. DFS is easiest.



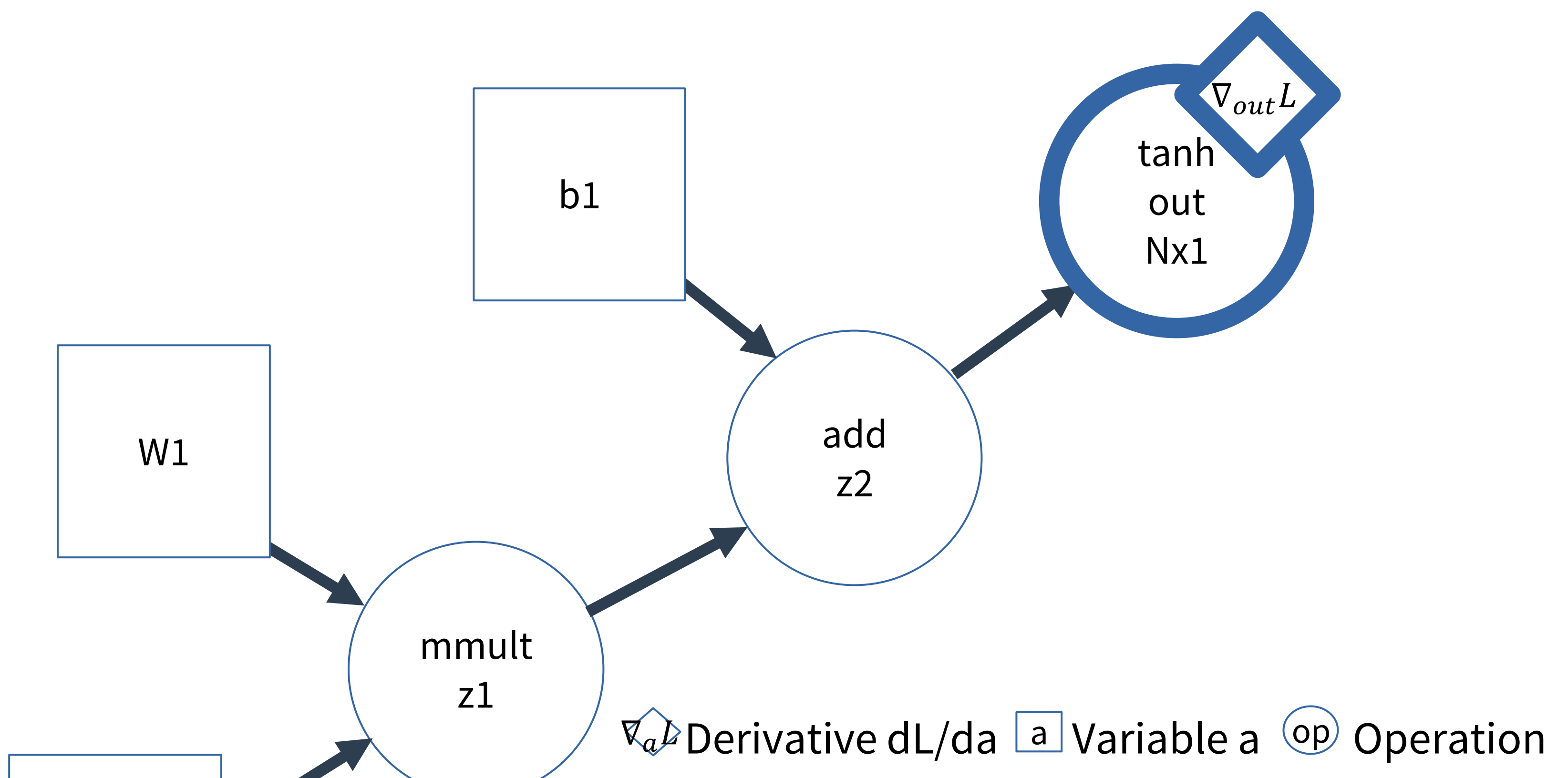
Simple MLP

Feel free to follow along on paper.



Simple MLP

If the output, tanh out is Nx1, what is the shape of $\nabla_{out}L$?



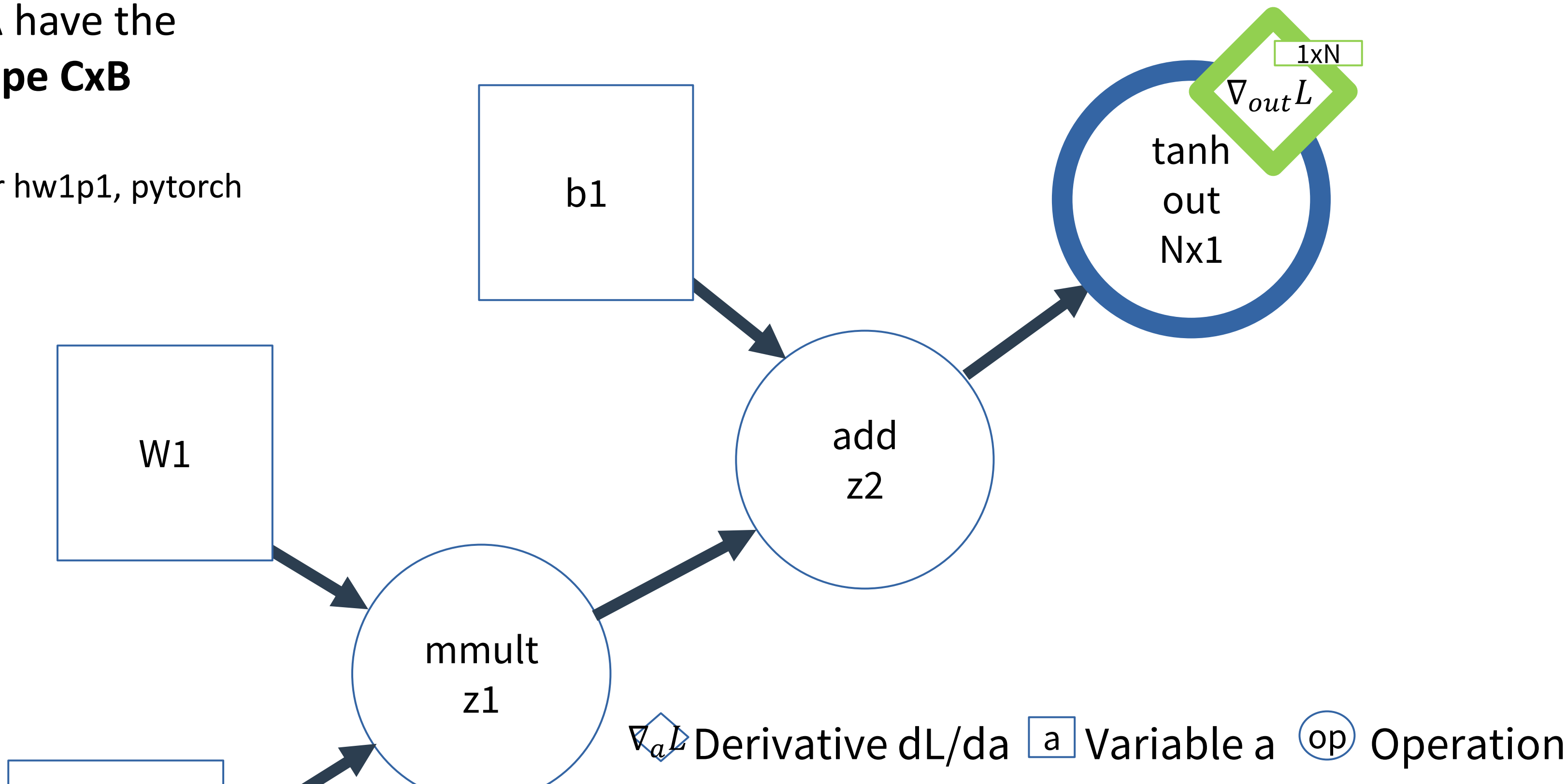
Simple MLP

If the output, tanh out is Nx1, what is

the shape of $\nabla_{out}L$?

If A has shape BxC,
gradients w.r.t. A have the
transpose shape CxB

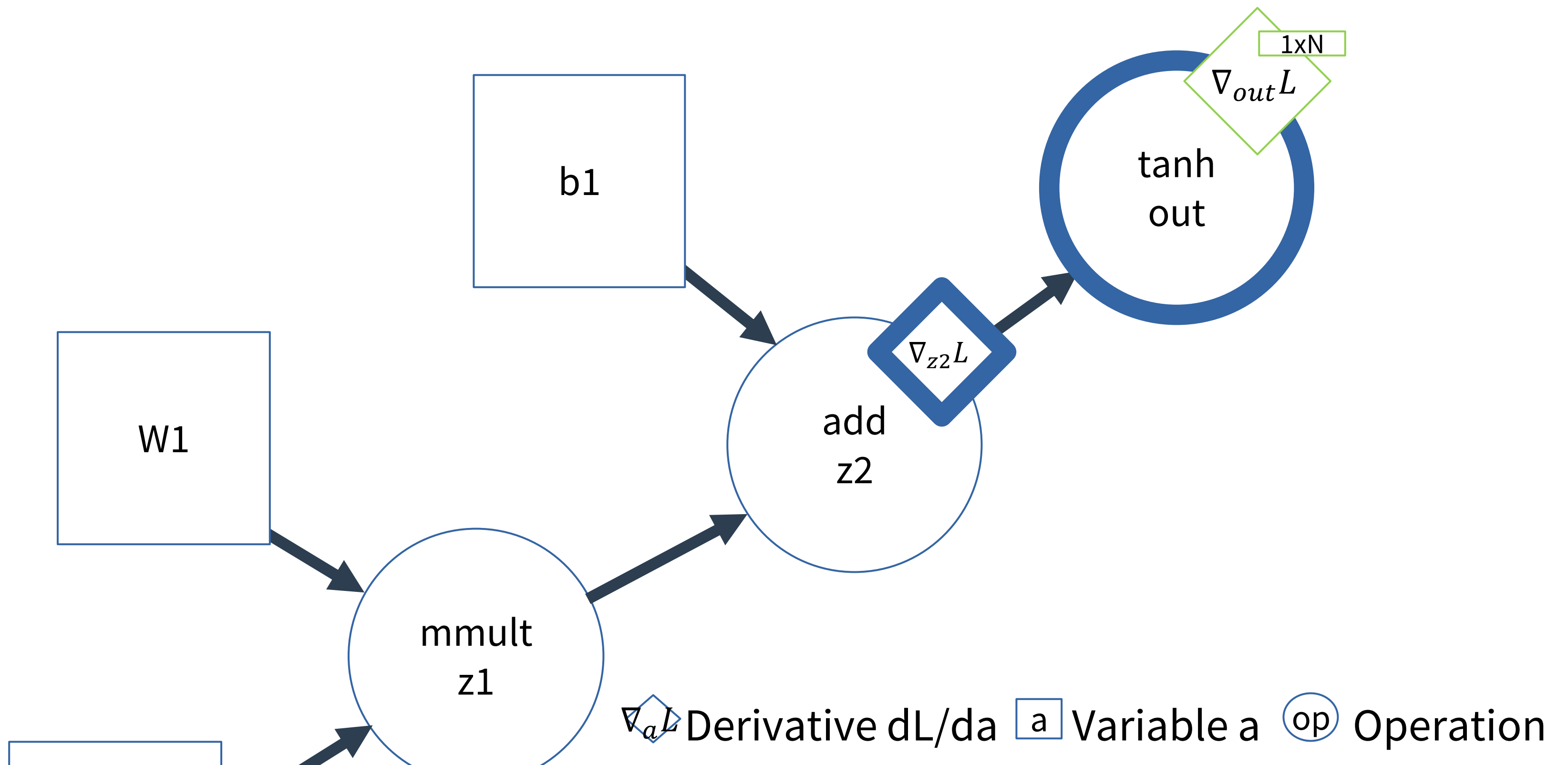
* shape isn't transpose for hw1p1, pytorch



Simple MLP

In terms of the chain rule, what is the backward function of tanh out ?

I.e., in terms of the chain rule, what is $\nabla_{z2}L$?



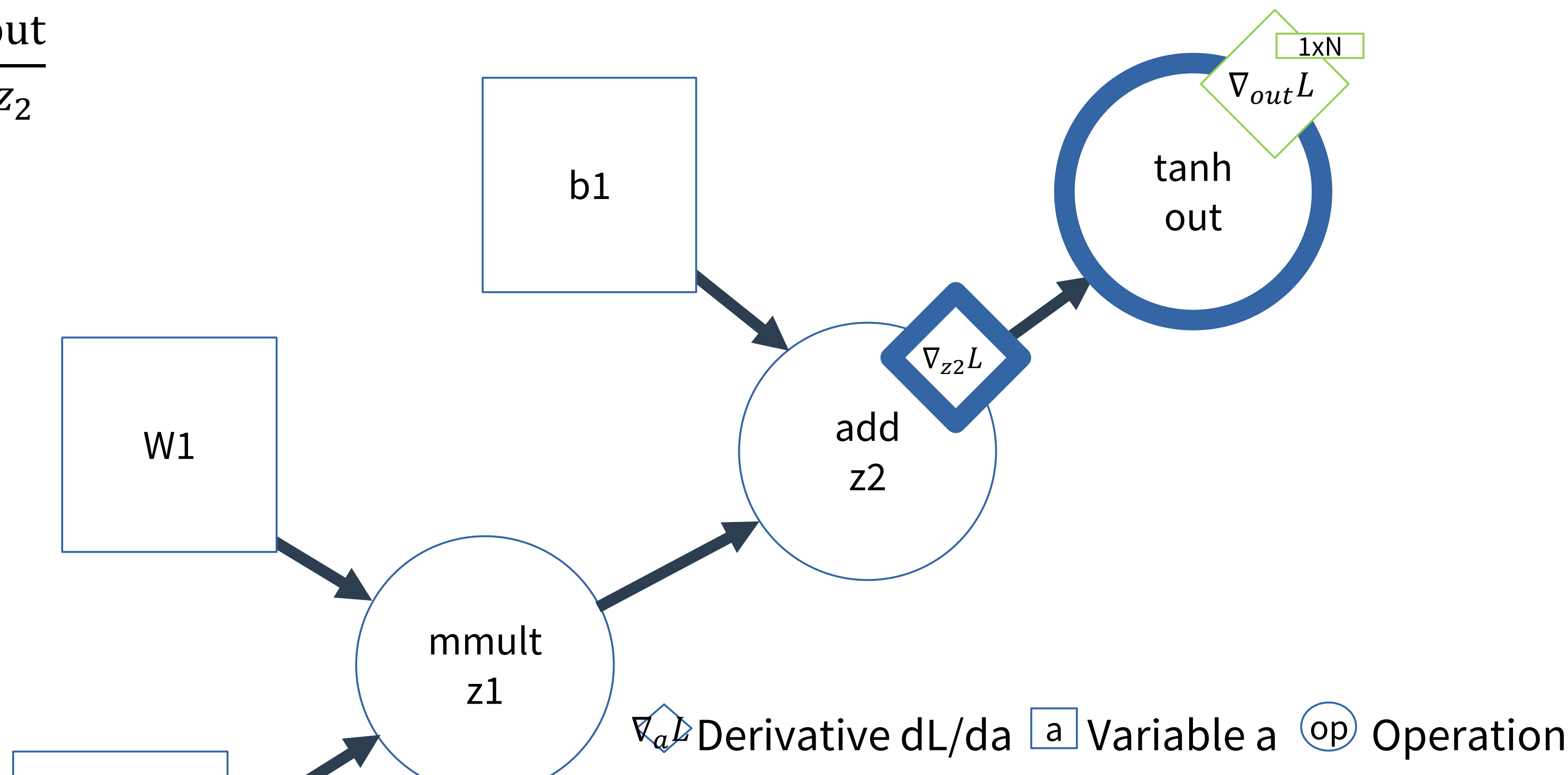
Simple MLP

In terms of the chain rule, what is the backward

function of tanh out ?

I.e., in terms of the chain rule, what is $\nabla_{z2}L$?

$$\frac{d\mathrm{L}}{dz_2} = \frac{d\mathrm{L}}{\mathrm{dout}} * \frac{\mathrm{dout}}{dz_2}$$

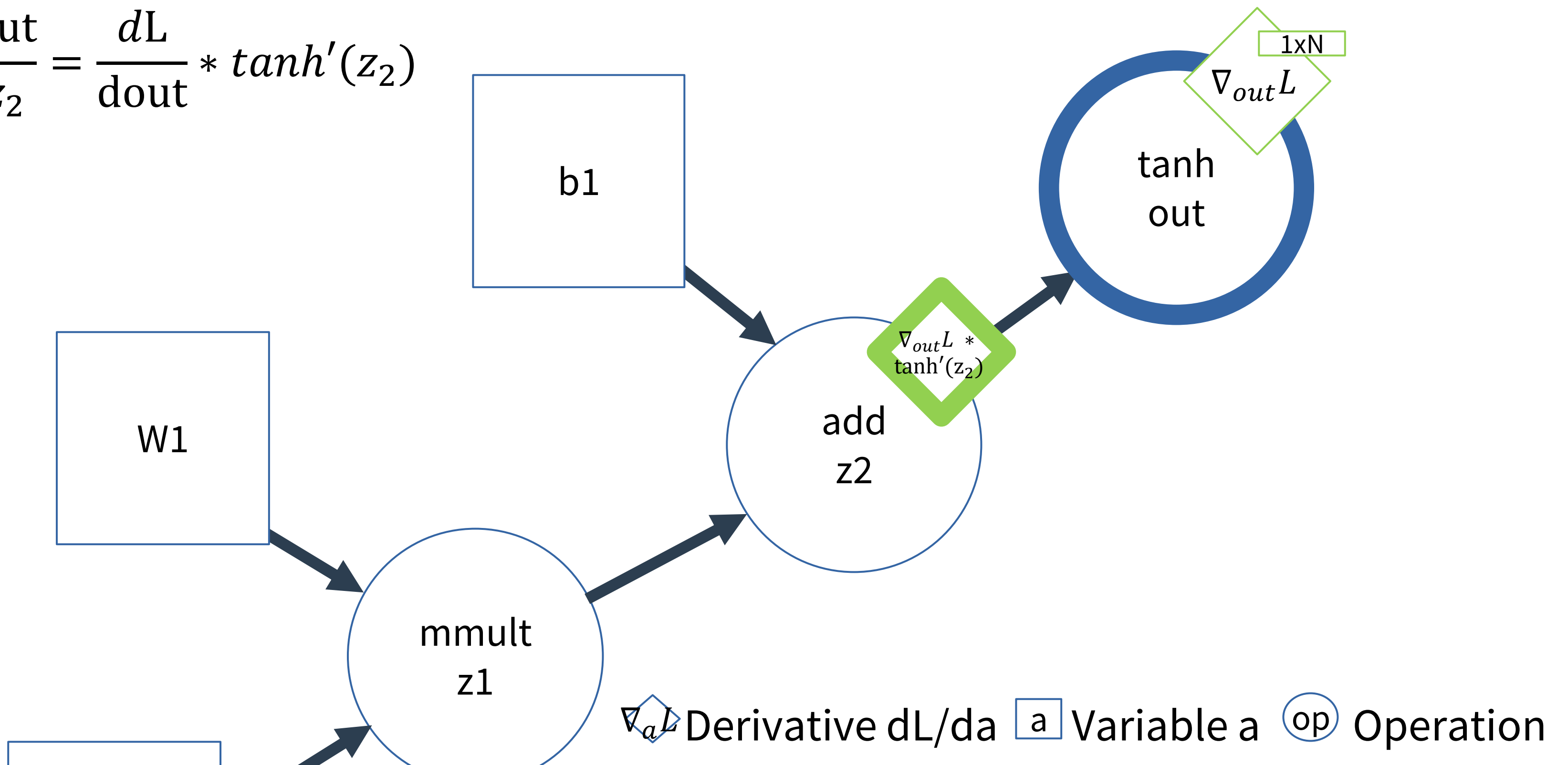


Simple MLP

In terms of the chain rule, what is the backward function of tanh out ?

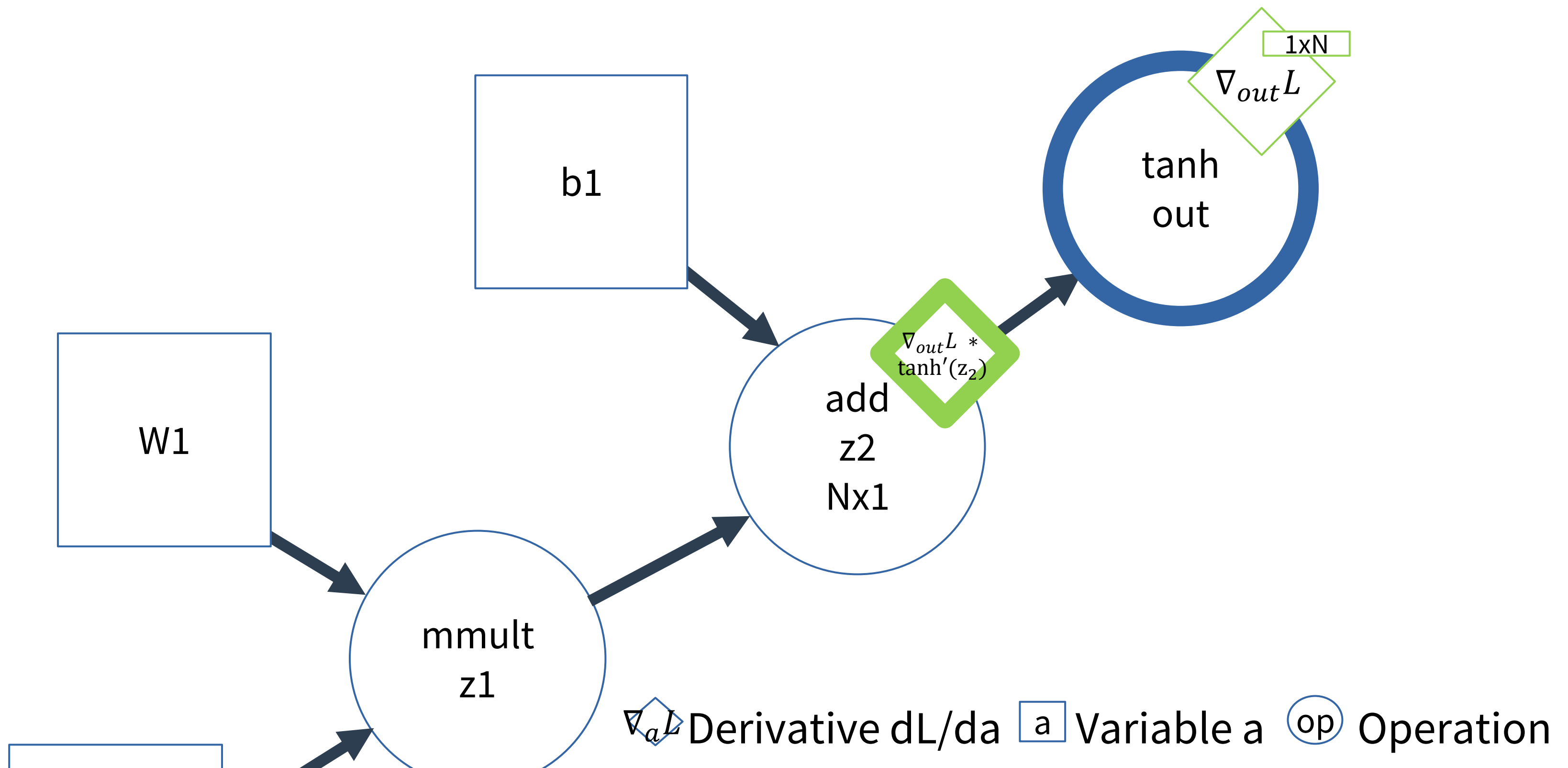
I.e., in terms of the chain rule, what is $\nabla_{z2}L$?

$$\frac{dL}{dz_2} = \frac{dL}{dout} * \frac{dout}{dz_2} = \frac{dL}{dout} * tanh'(z_2)$$



Simple MLP

What is the shape of $\nabla_{z2}L$?

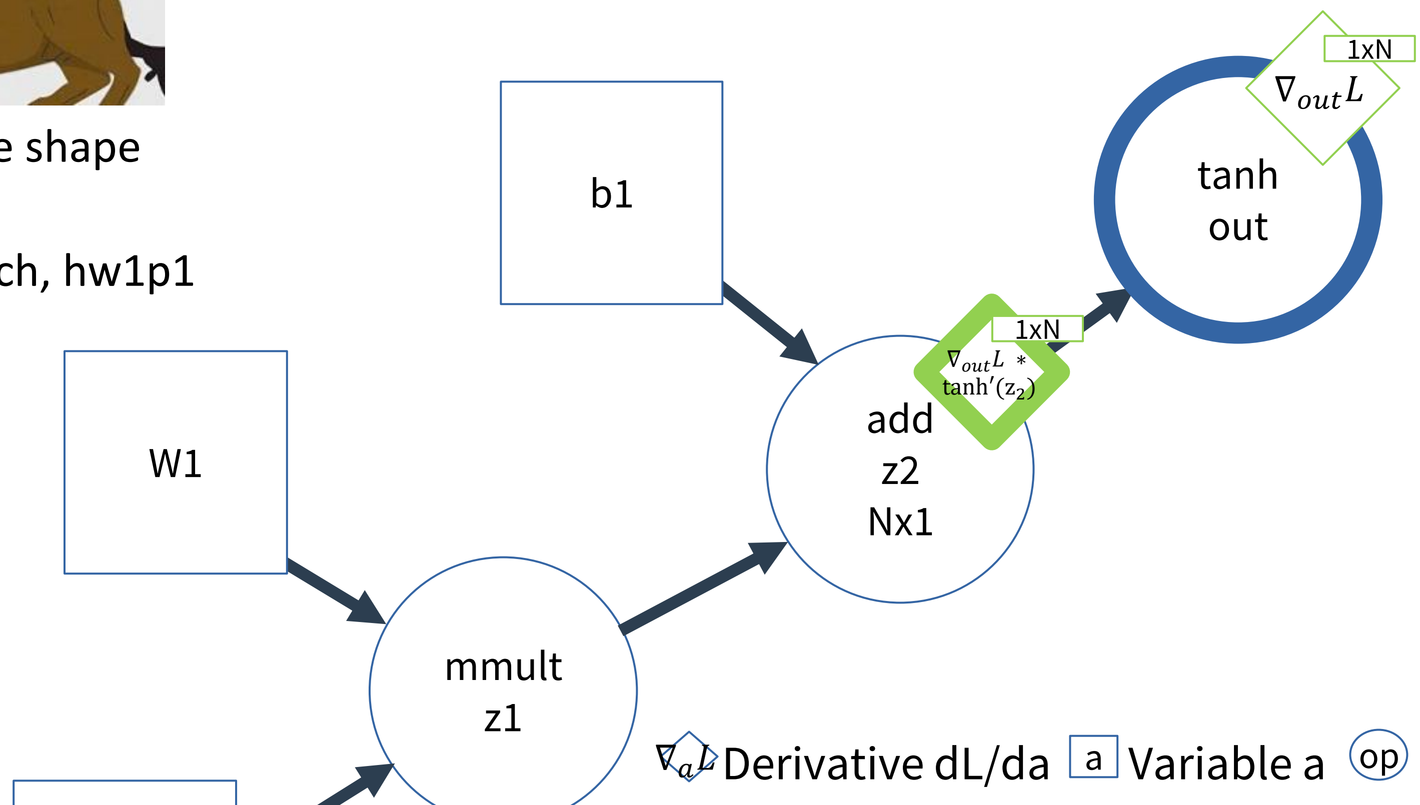


Simple MLP

What is the shape of $\nabla_{z2}L$?

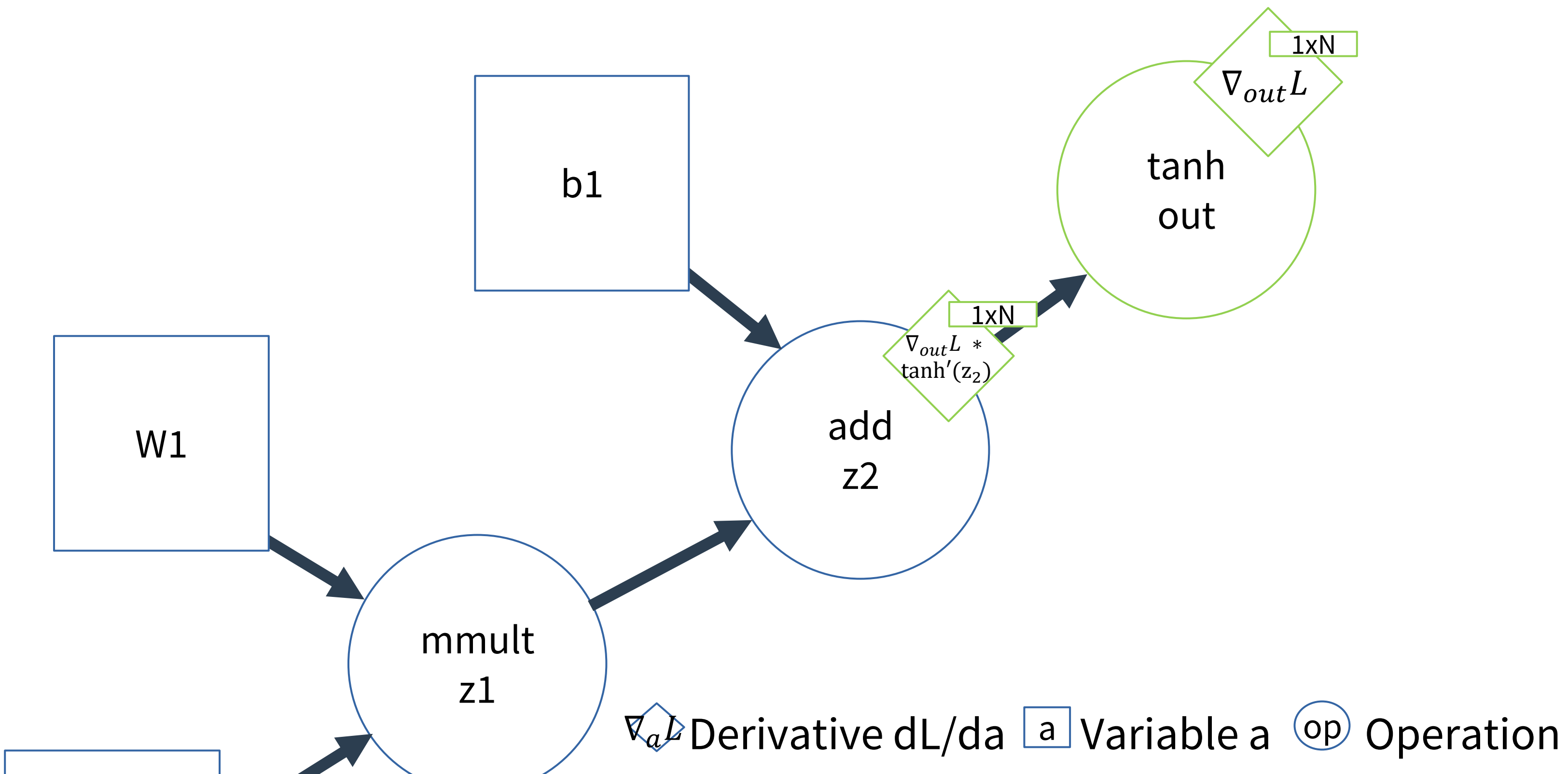
The transpose shape

*except in pytorch, hw1p1



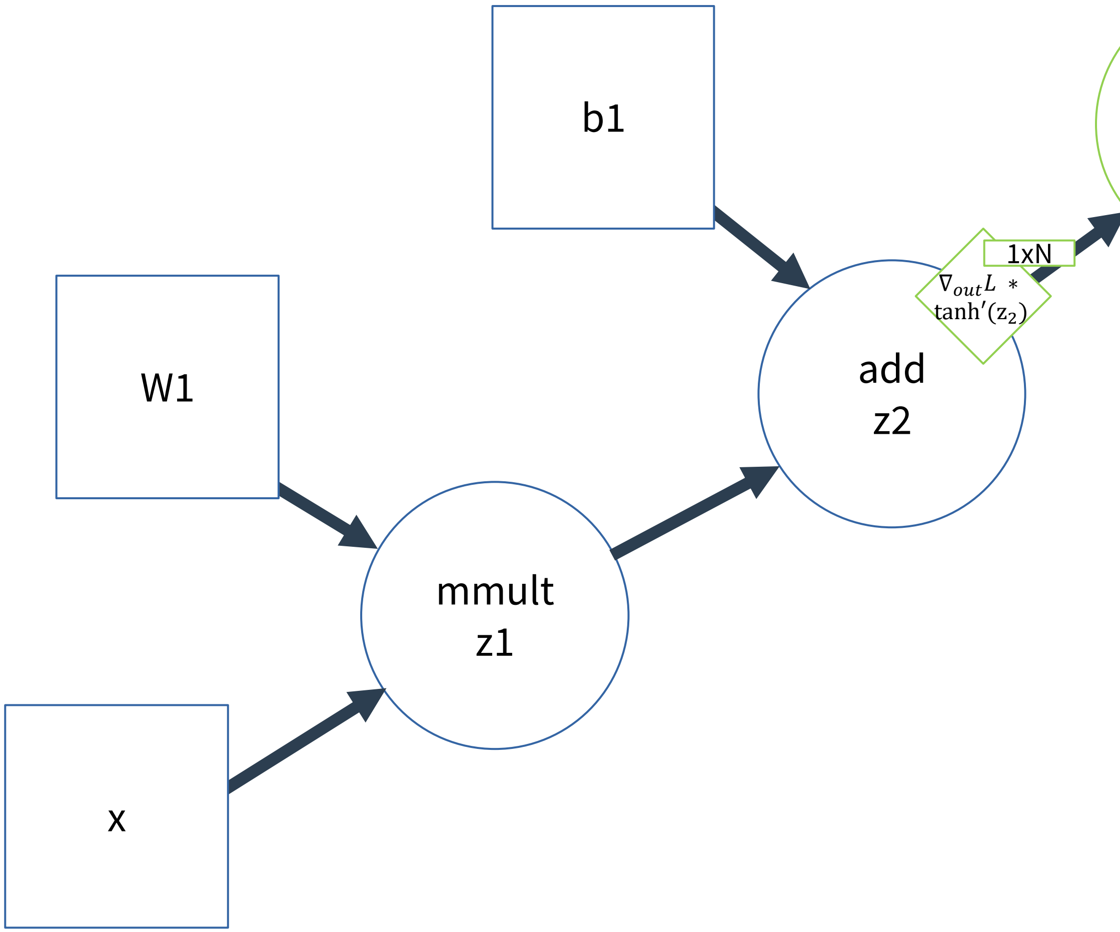
$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP



b1
W1
mmult
z1
add
z2
tanh
out
1xN
$\nabla_{out} L$
1xN
$\nabla_{out} L$ *
$\tanh'(z_2)$
$\nabla_a L$ Derivative dL/da
a Variable a
op Operation

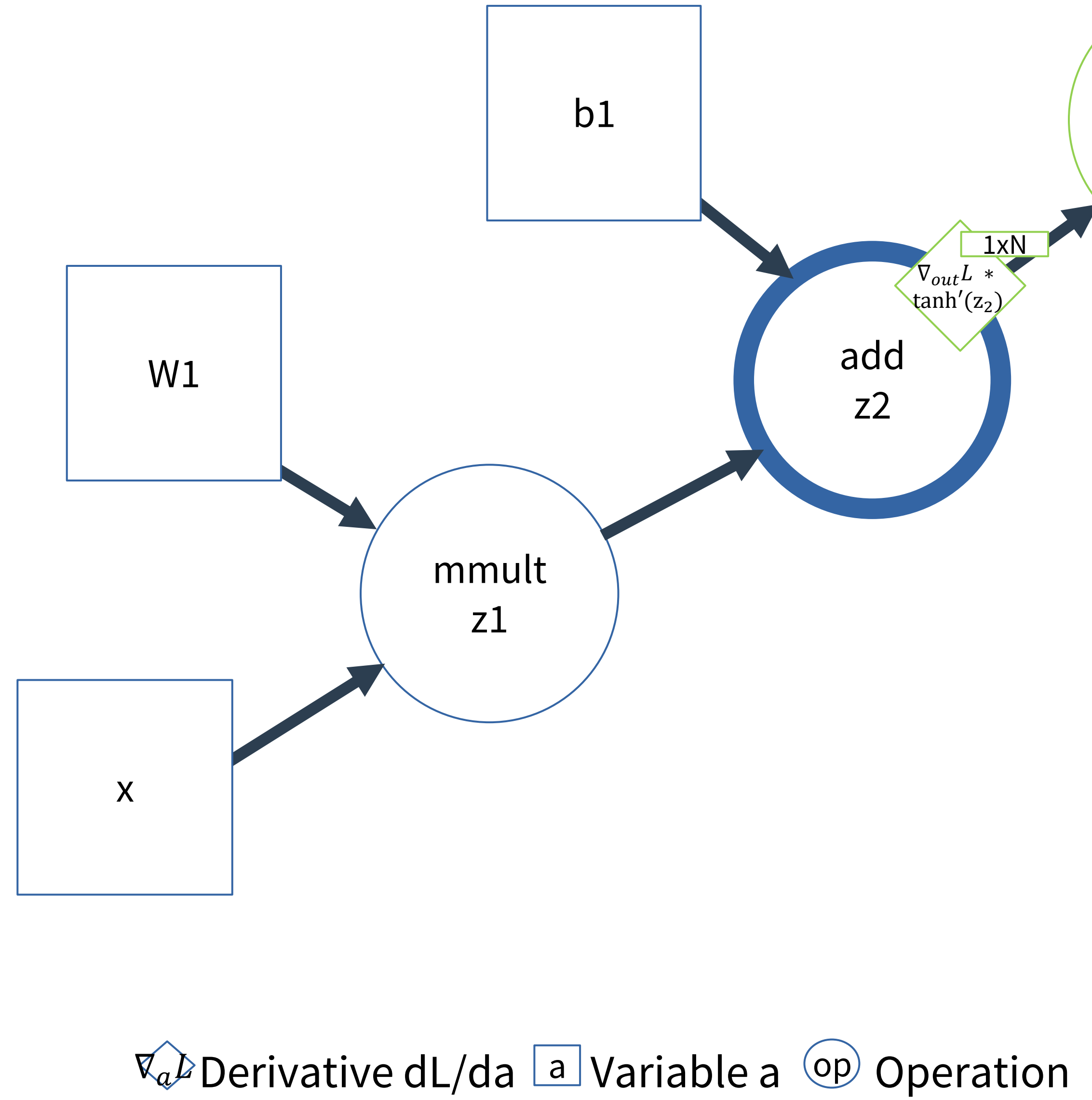
Simple MLP



$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

We will continue the graph search by visiting add z2.

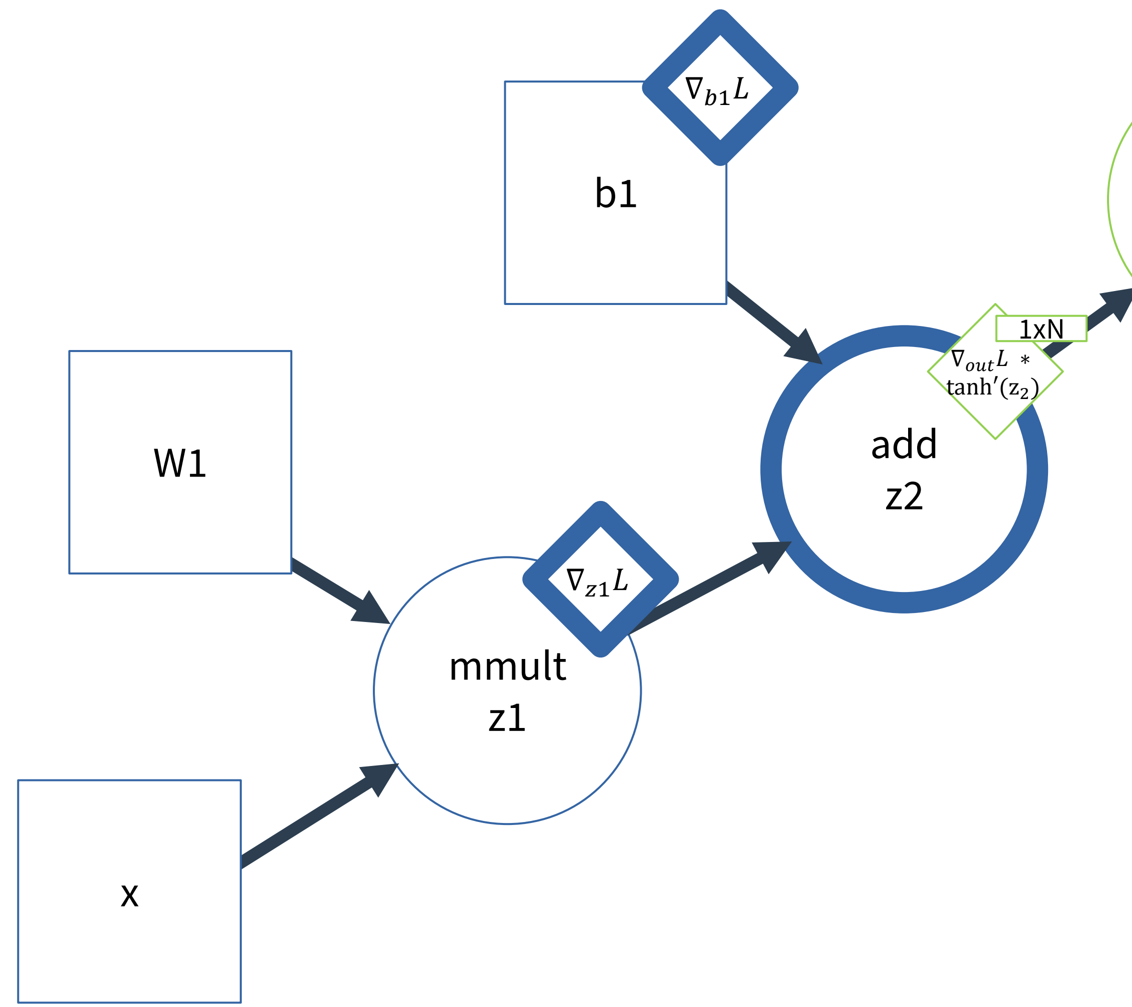


Simple MLP

What is the backward function of add z2 ?

I.e., what are $\nabla_{b1}L$ and $\nabla_{z1}L$?

Hint: $c = a + b, \quad \frac{dL}{da} = \frac{dL}{dc}\frac{dc}{da}$



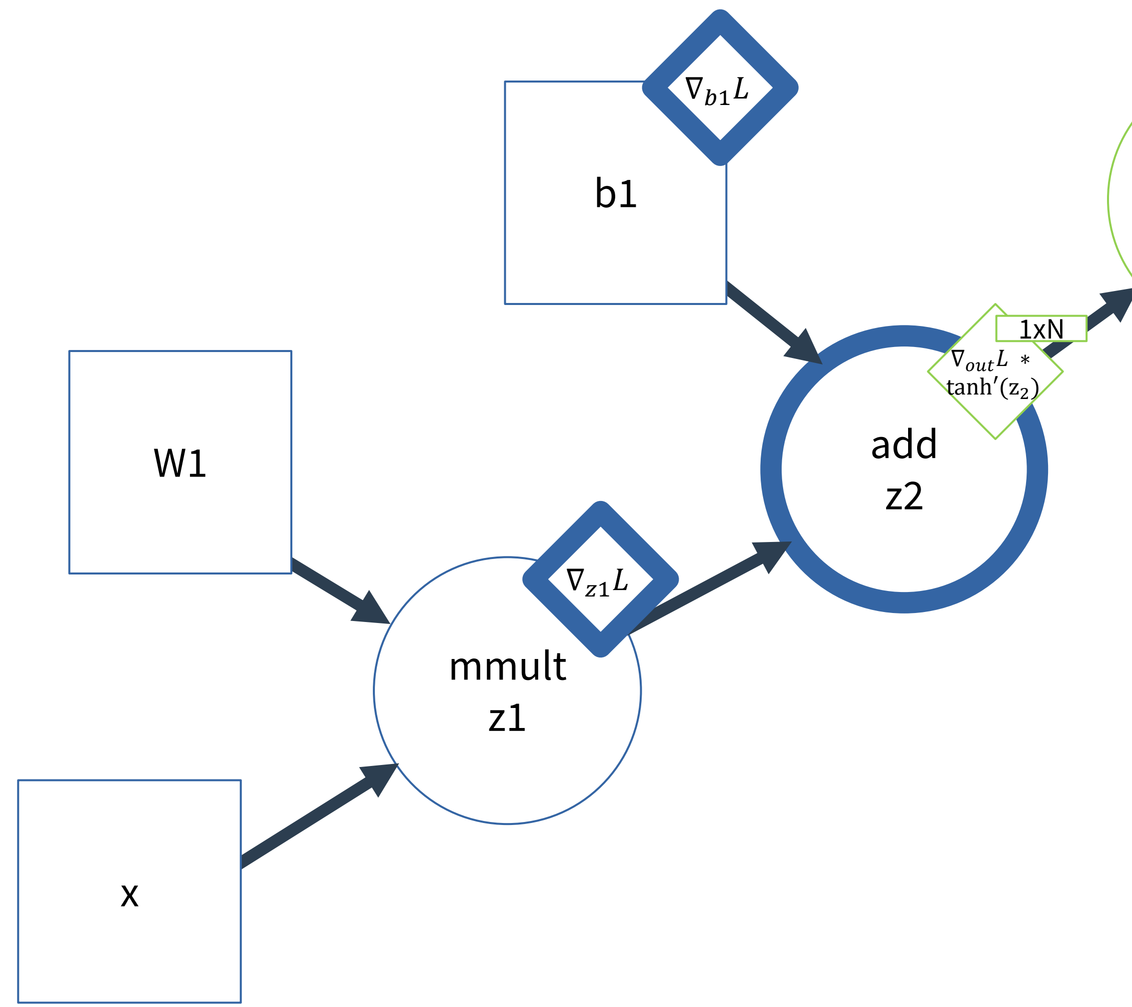
$\nabla_a L$ Derivative dL/da   a Variable a   op Operation

Simple MLP

What is the backward function of add z2 ?

I.e., what are $\nabla_{b1}L$ and $\nabla_{z1}L$?

Hint: $c = a + b$, $\frac{dL}{da} = \frac{dL}{dc}\cancel{\frac{dc}{da}}$



$\nabla_a L$ Derivative dL/da a Variable a op Operation

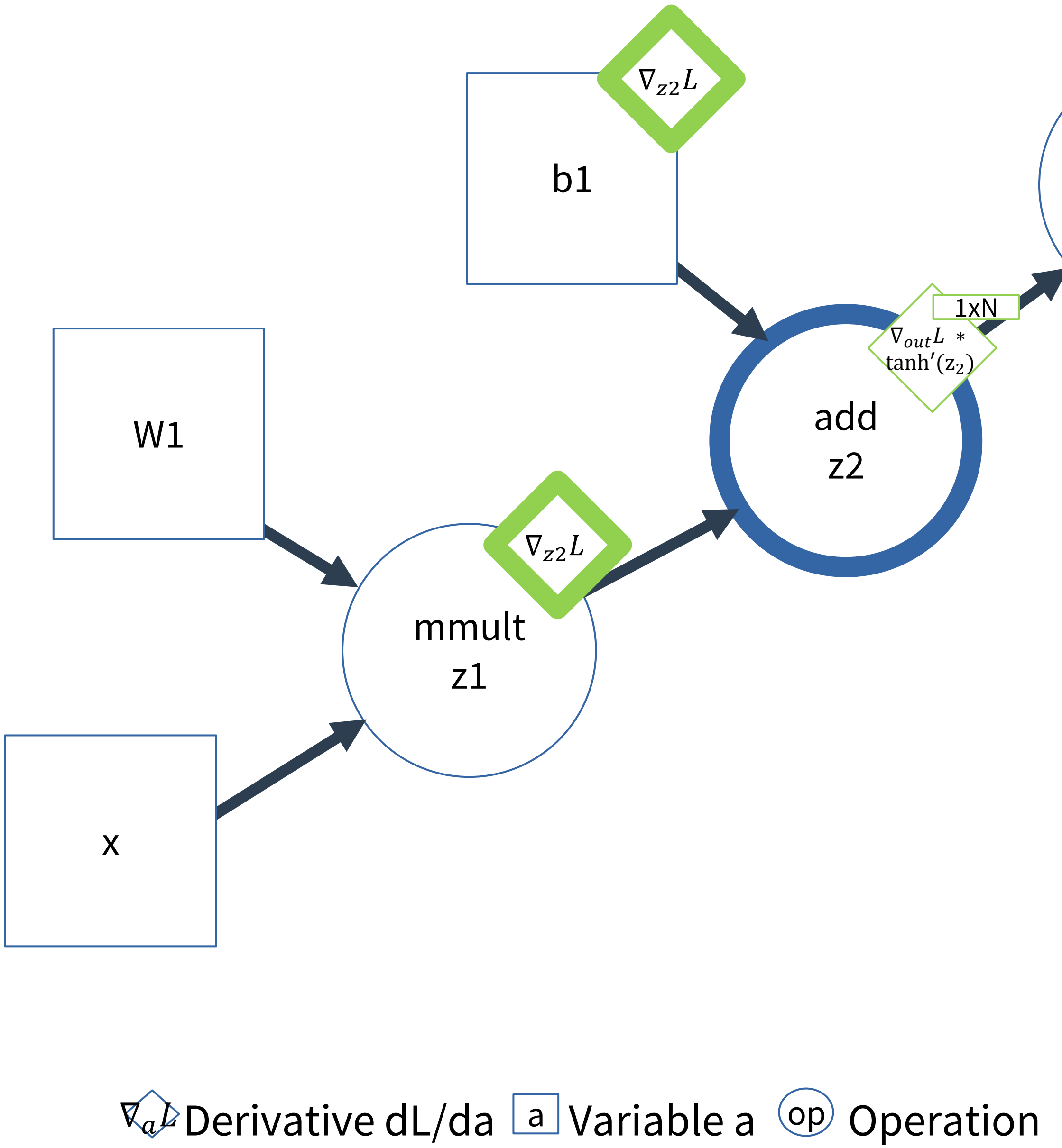
Simple MLP

What is the backward function of add z2 ?

I.e., what are $\nabla_{b1}L$ and $\nabla_{z1}L$?

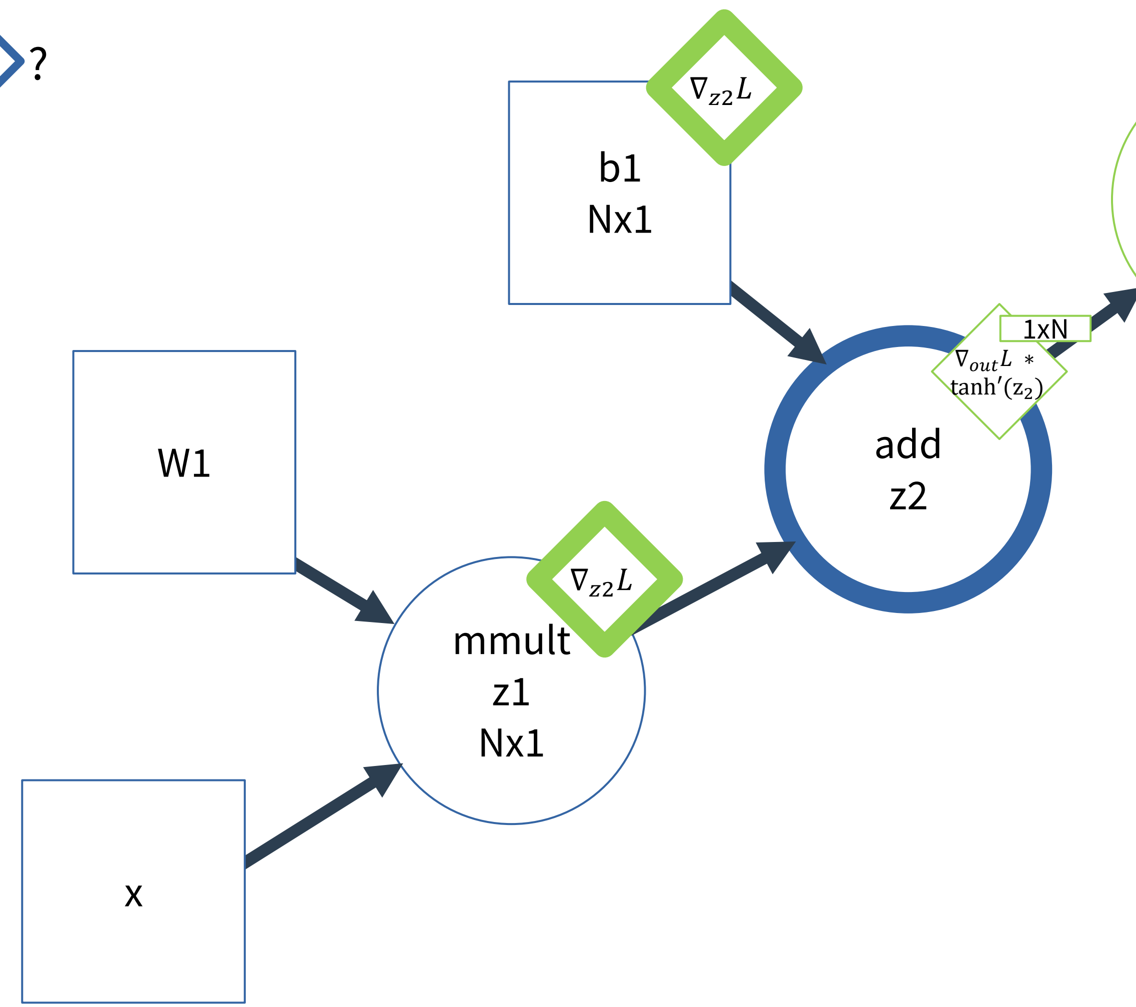
Hint: $c = a + b$, $\frac{dL}{da} = \frac{dL}{dc}\cancel{\frac{dc}{da}}$

$$\frac{dL}{db_1} = \frac{dL}{dz_2}, \qquad \frac{dL}{dz_1} = \frac{dL}{dz_2}$$



Simple MLP

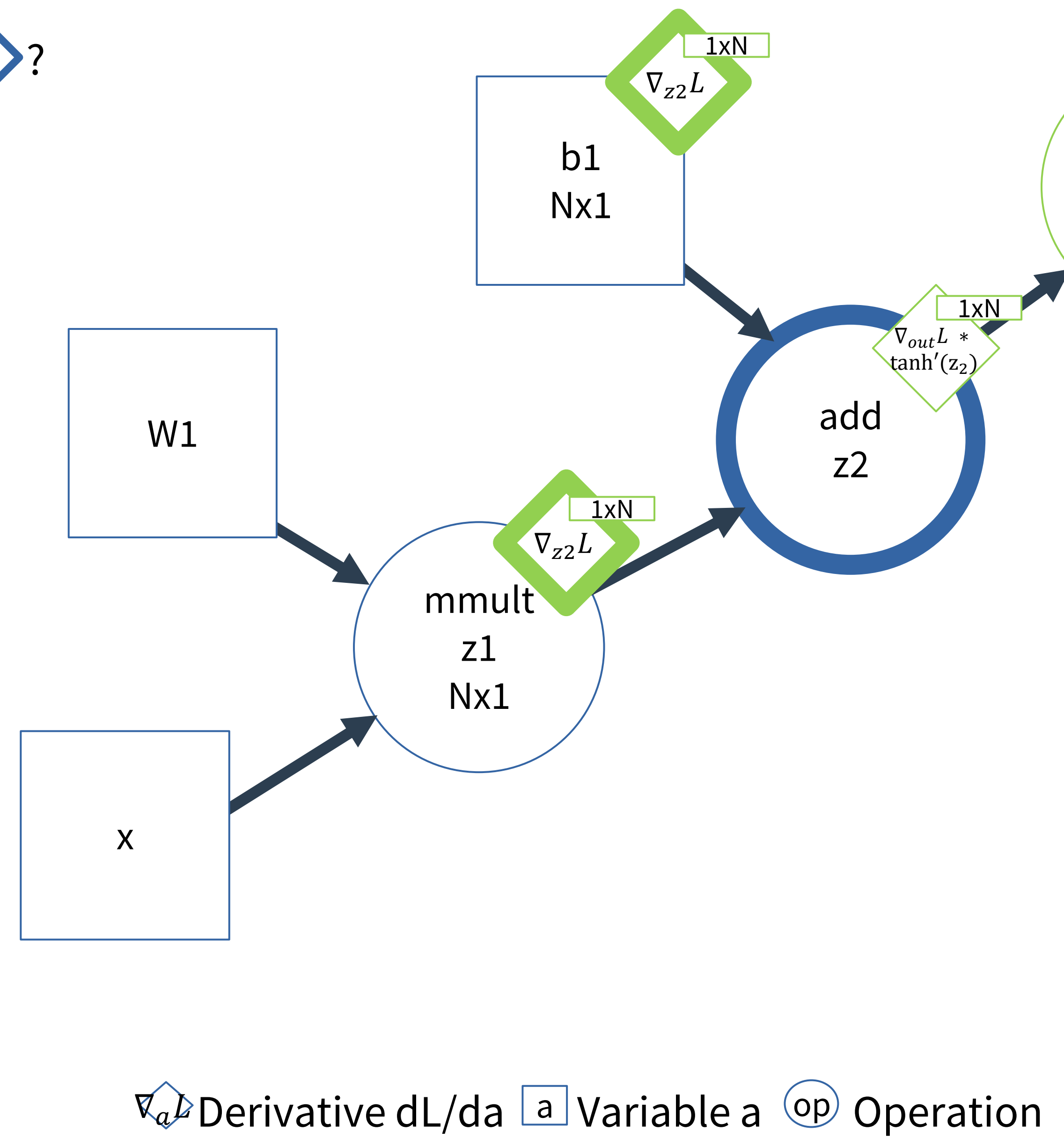
What are the shapes of $\nabla_{b1}L$ and $\nabla_{z1}L$?



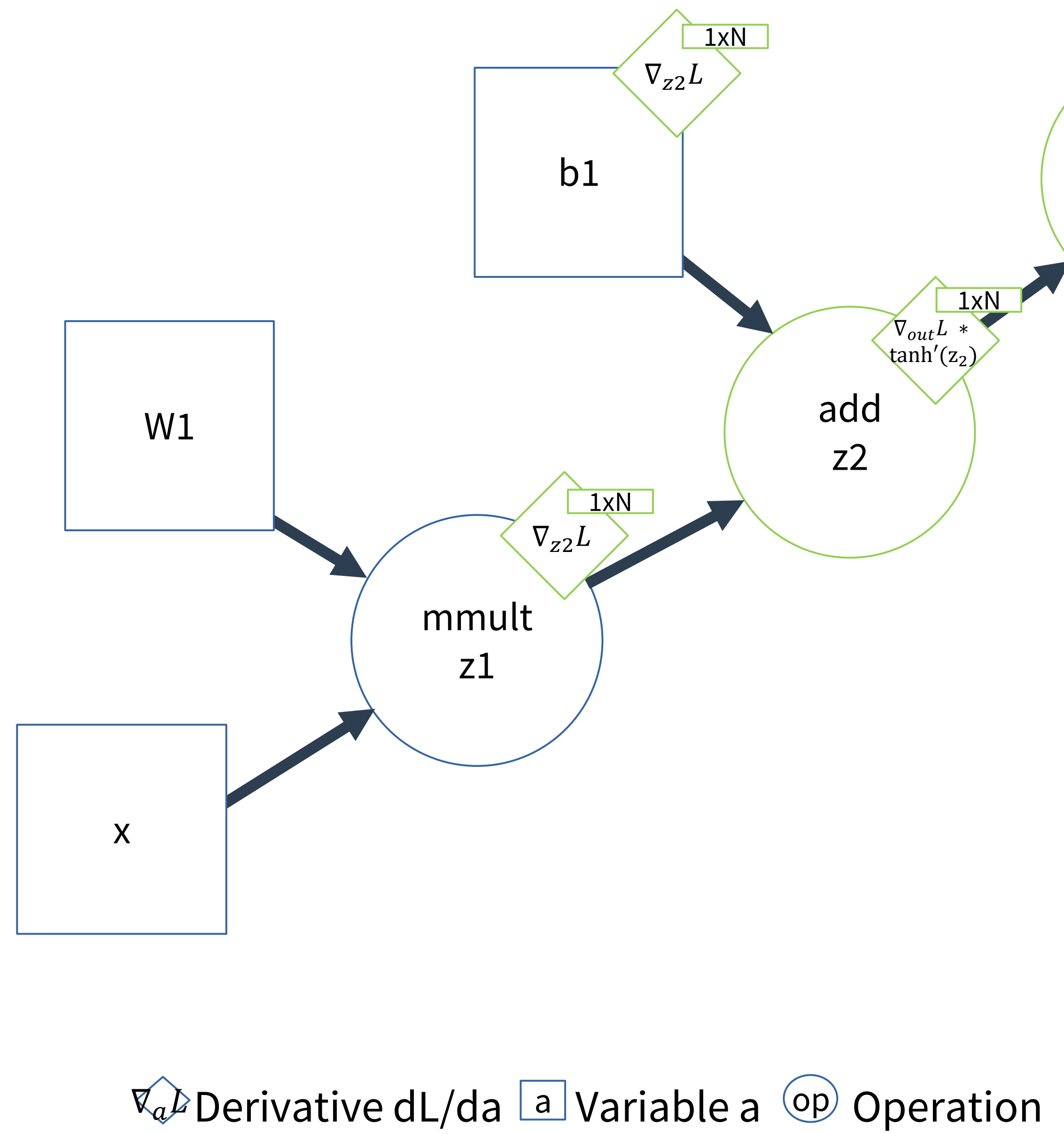
$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

What are the shapes of $\nabla_{b1}L$ and $\nabla_{z1}L$?



Simple MLP

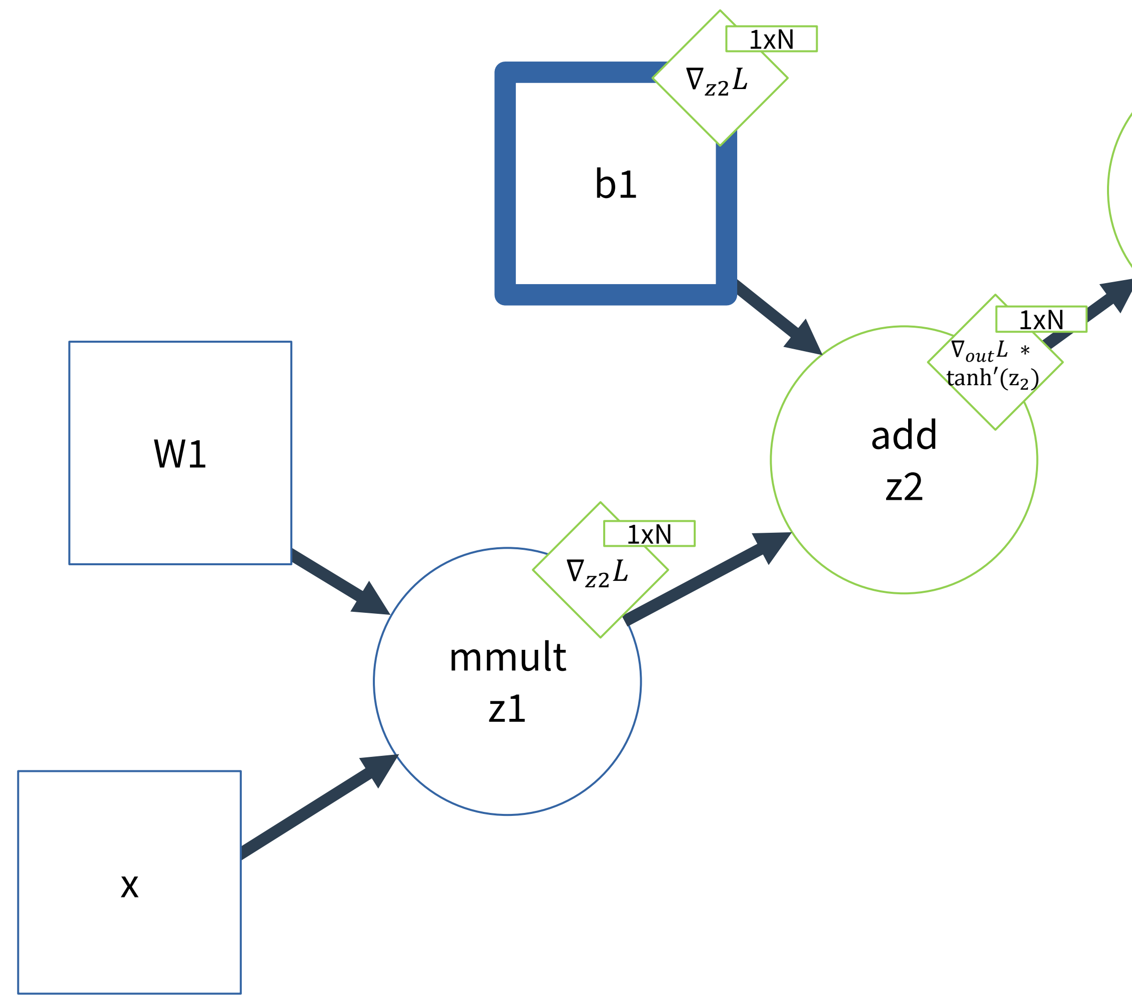


1xN
$\nabla_{z2}L$
b1
1xN
$\nabla_{out}L$ *
tanh'(z_2)
W1
add
z2
1xN
$\nabla_{z2}L$
mmult
z1
x
$\nabla_a L$ Derivative dL/da
a Variable a
op Operation

Simple MLP

We will continue the graph search by

visiting $\boxed{\text{b1}}$.

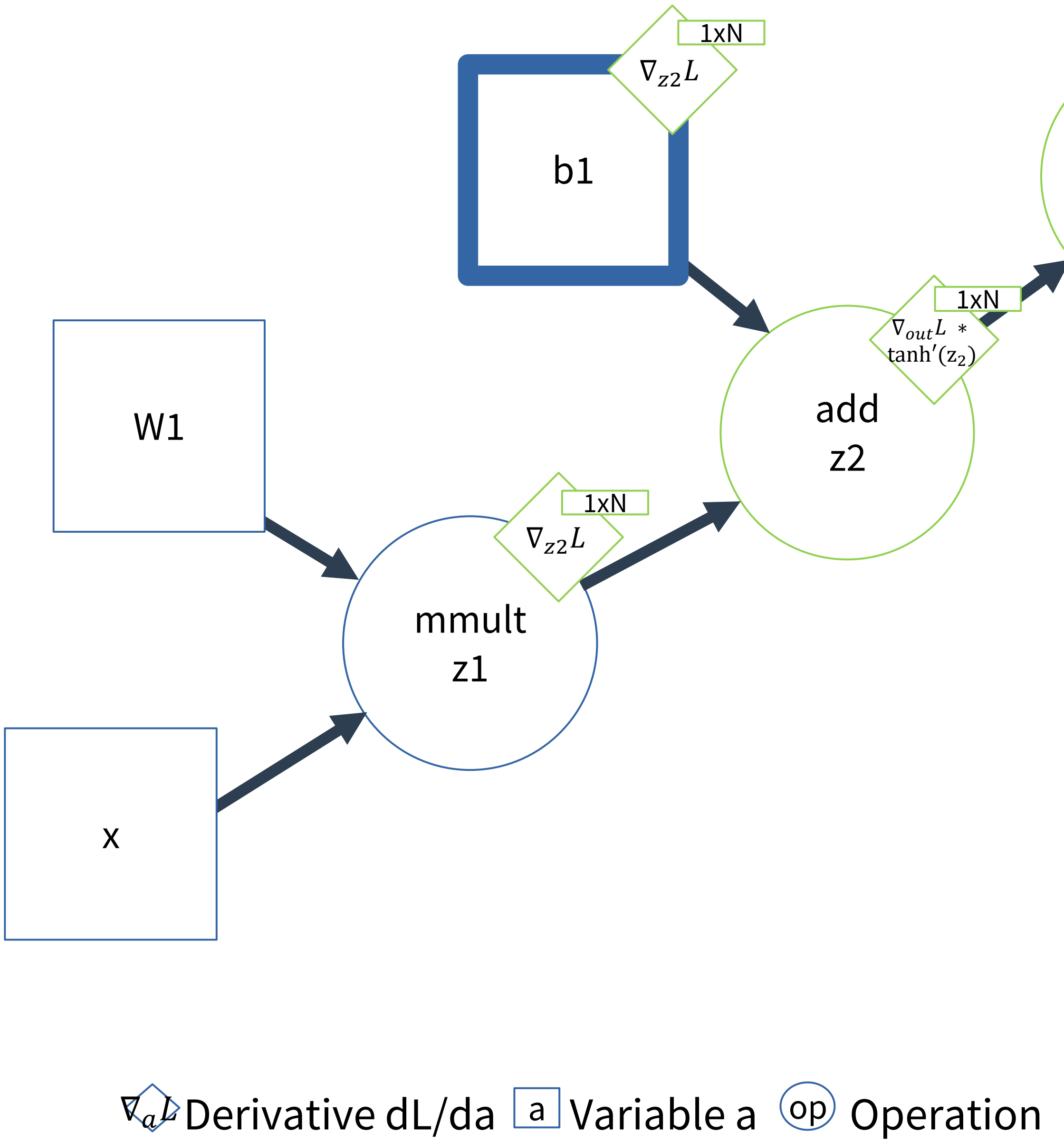


$\nabla_a L$ Derivative dL/da $\boxed{a}$ Variable a (op) Operation

Simple MLP

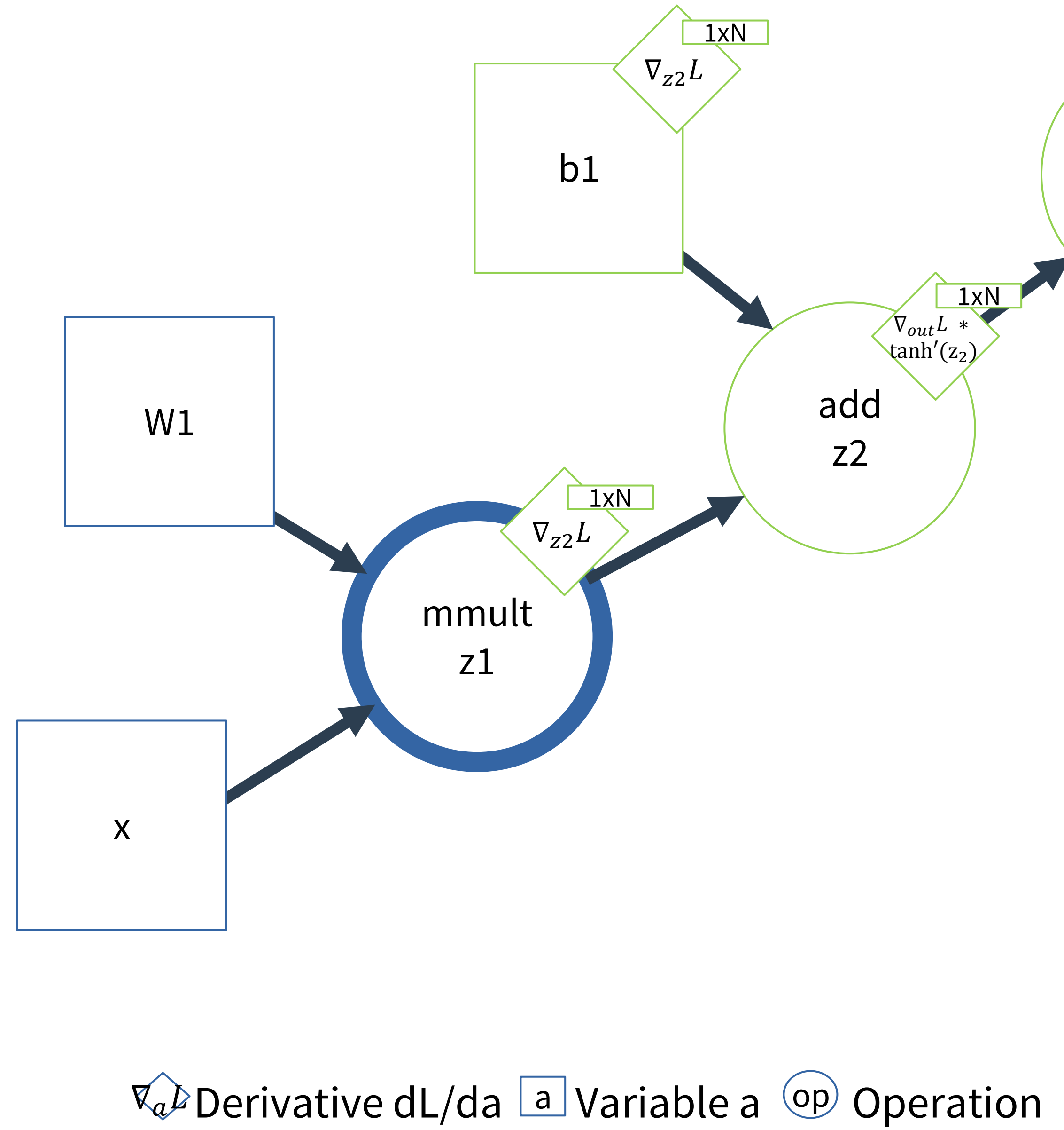
$\boxed{\text{b1}}$ has no gradient-enabled parents,

and we want it's gradient, so its backward function is to **accumulate** (i.e. save) the gradient passed to it.



Simple MLP

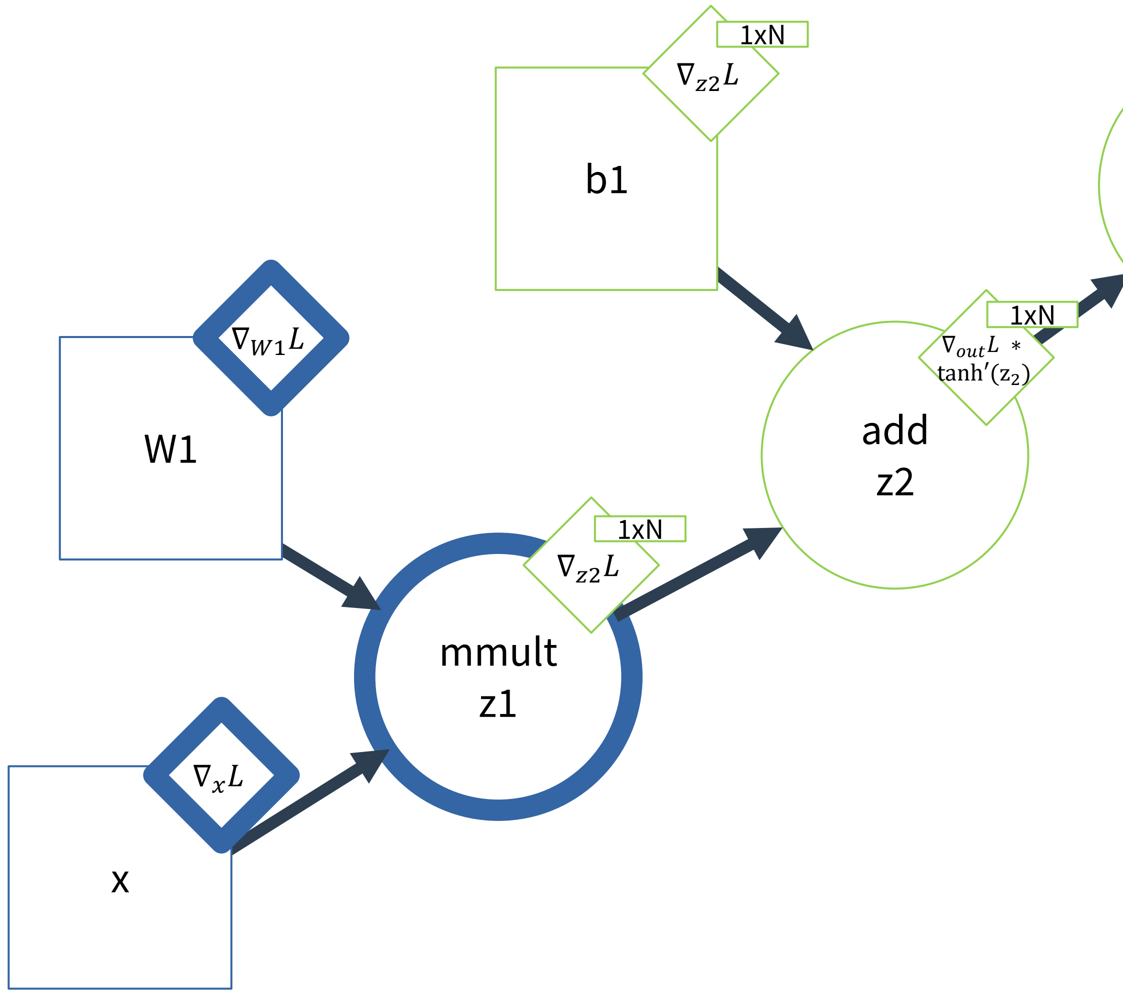
We will continue the graph search by visiting mmult z1.



Simple MLP

What is the backward function of mmult z1 ?

I.e., what are $\nabla_{W1}L$ and $\nabla_{x}L$?



$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

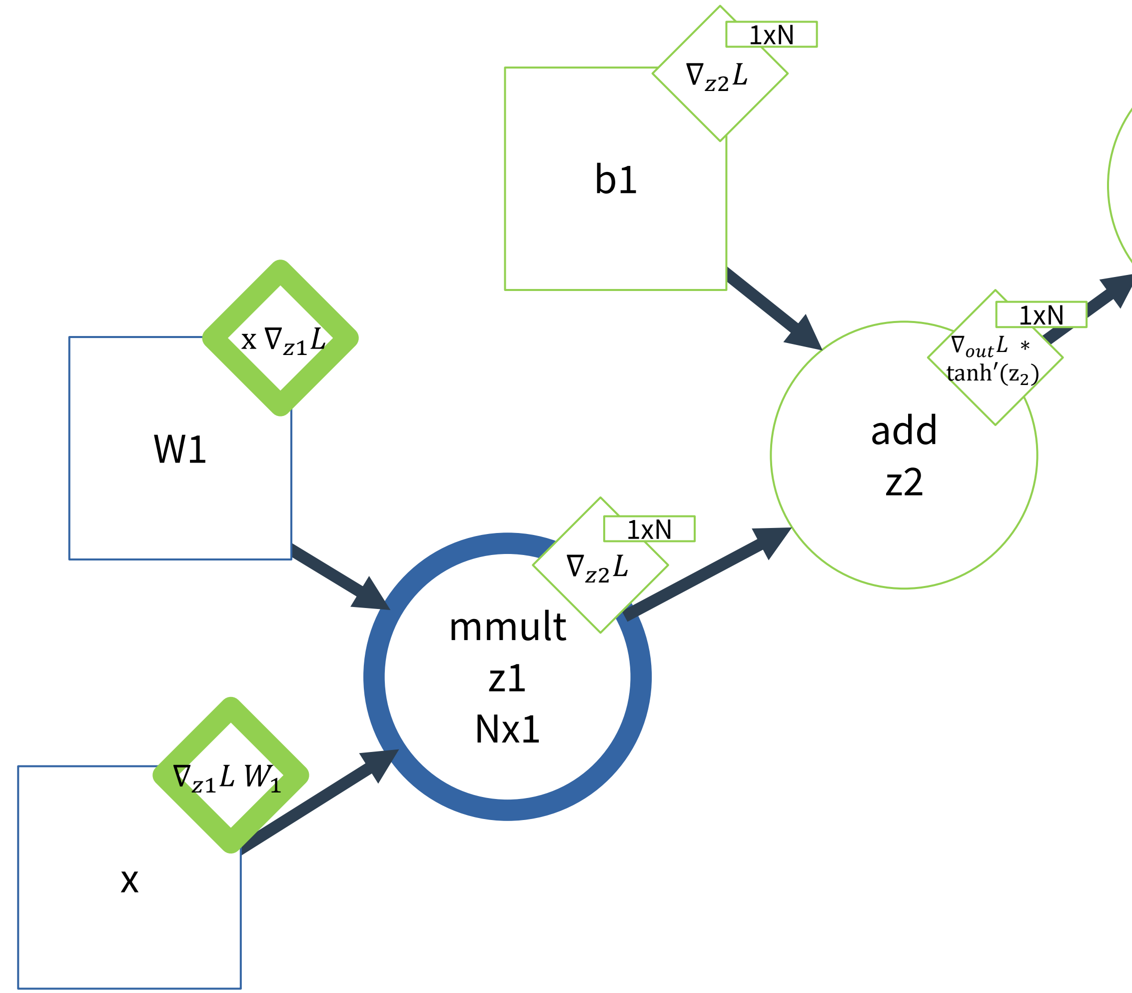
What is the backward function of mmult z1 ?

I.e., what are $\nabla_{W1}L$ and $\nabla_{x}L$?

Given matrix $\nabla_{AB}L$:

$$\nabla_A L = B\ \nabla_{AB} L$$
$$\nabla_B L = \nabla_{AB} L\ A$$

These are matrix multiplies.
Confirm this rule for yourself!



$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

What is the backward function of mmult z1 ?

I.e., what are $\nabla_{W1} L$ and $\nabla_x L$?

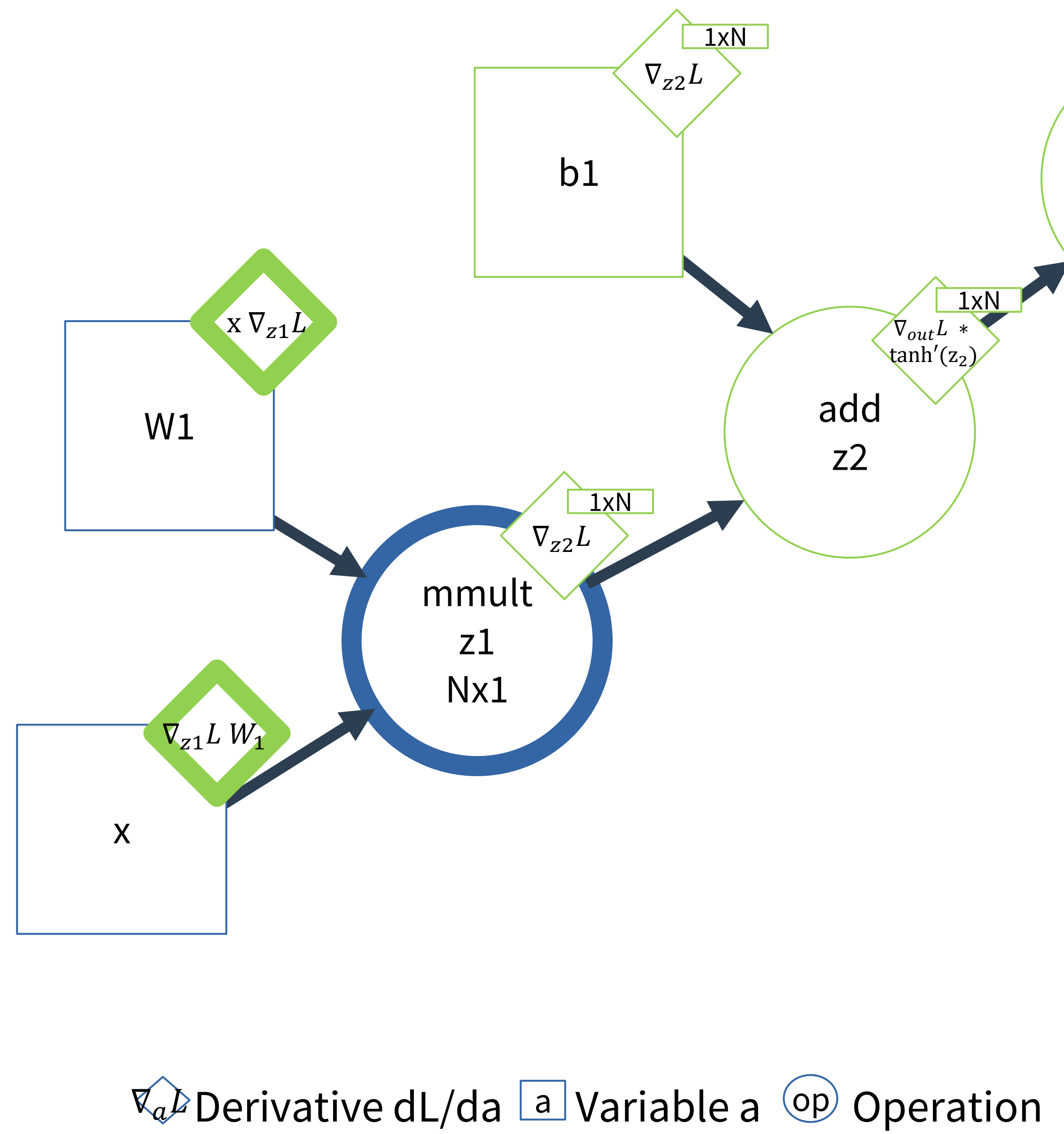
Given matrix $\nabla_{AB} L$:

$$\nabla_A L = B\ \nabla_{AB} L$$
$$\nabla_B L = \nabla_{AB} L\ A$$

These are matrix multiplies.
Confirm this rule for yourself!

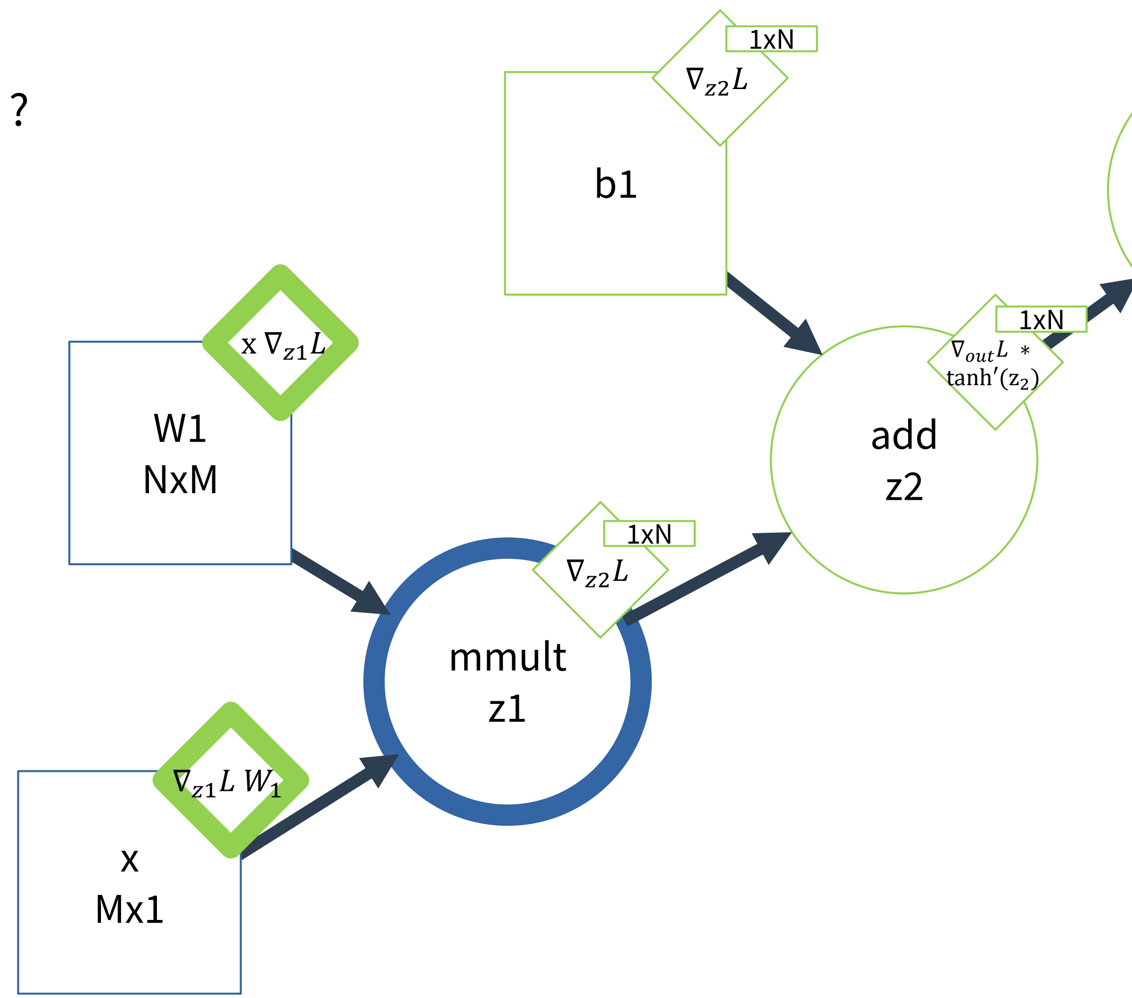
When the gradient isn't transposed (1p1):

$$\nabla_A L = \nabla_{AB} L\ B^T$$
$$\nabla_B L = A^T\ \nabla_{AB} L$$



Simple MLP

What are the shapes of $\nabla_{W1}L$ and $\nabla_{x}L$?

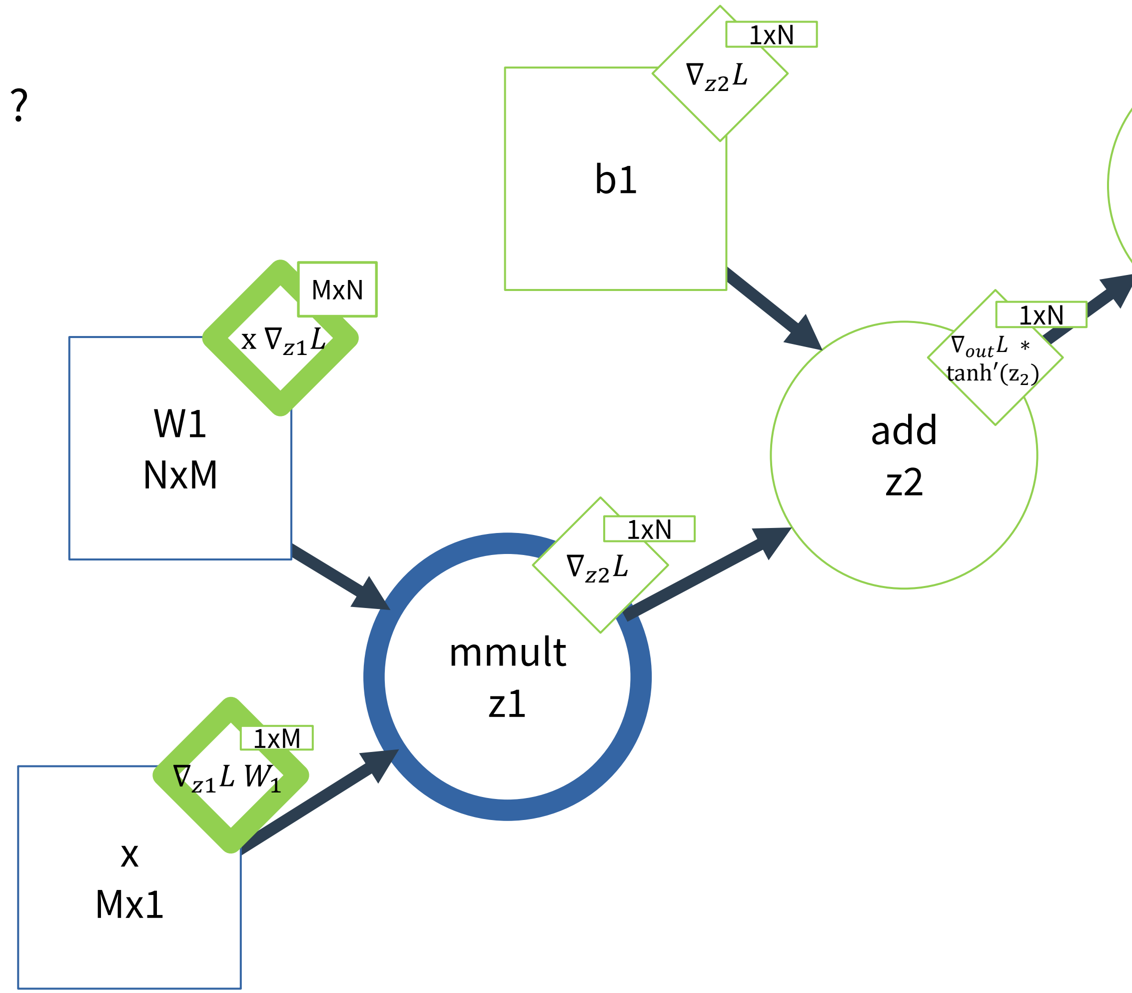


$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

What are the shapes of $\nabla_{W1} L$ and $\nabla_x L$?

... **transpose** except in hw1p1, pytorch ...

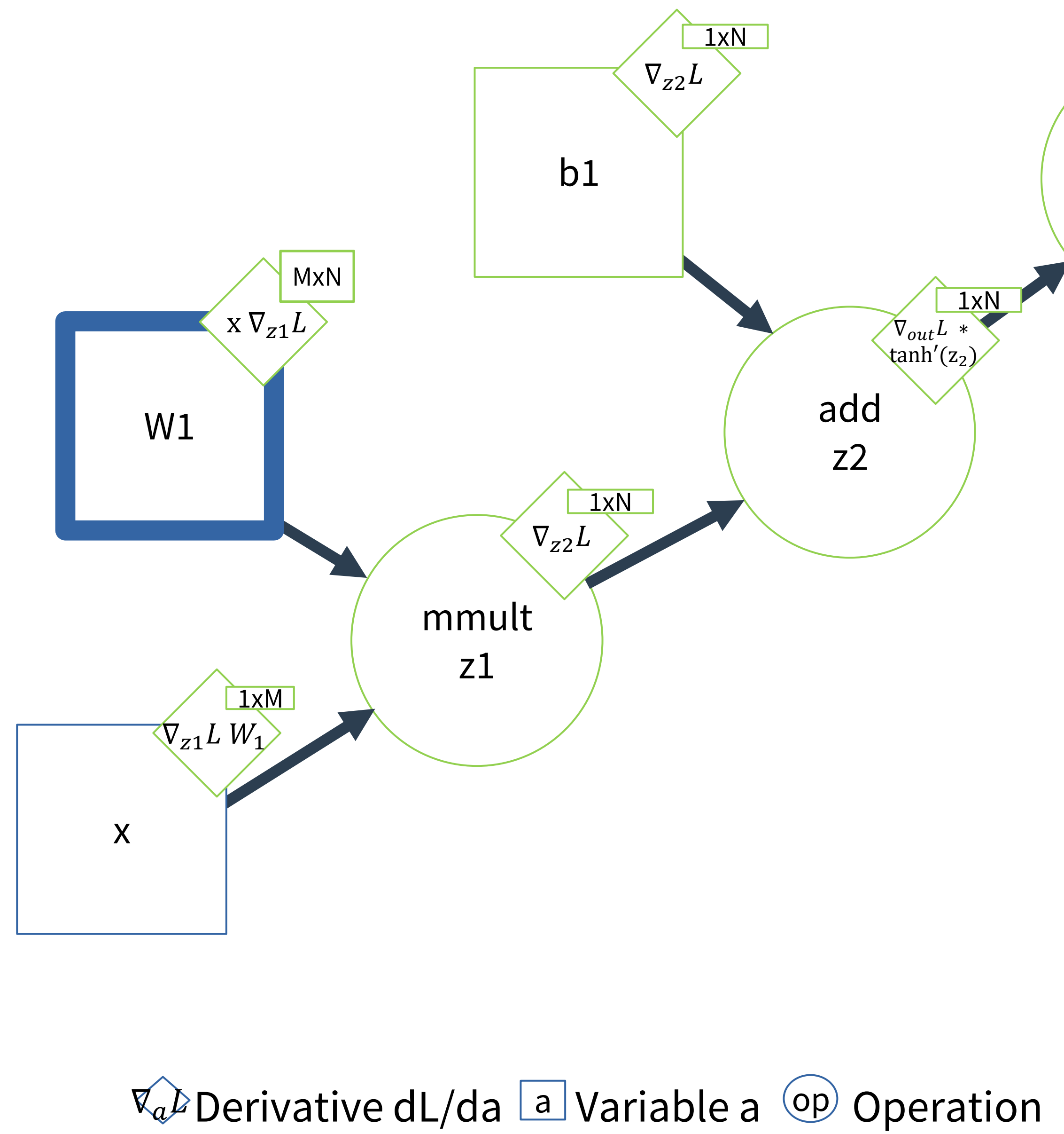


$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

We will continue the graph search by

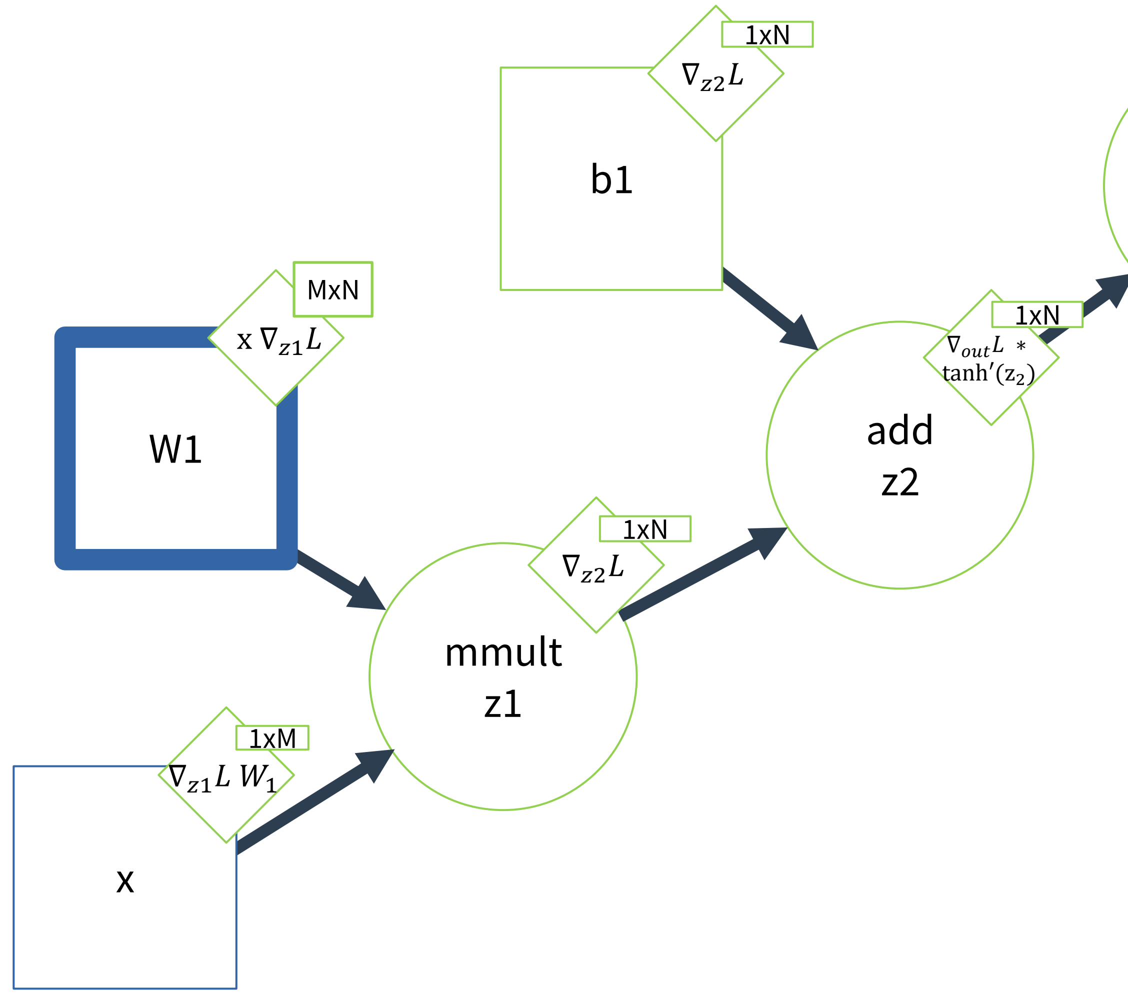
visiting $\boxed{\text{W1}}$.



Simple MLP

W1 has no gradient-enabled parents,

and we want it's gradient, so its backward function is to **accumulate** (i.e. save) the gradient passed to it.

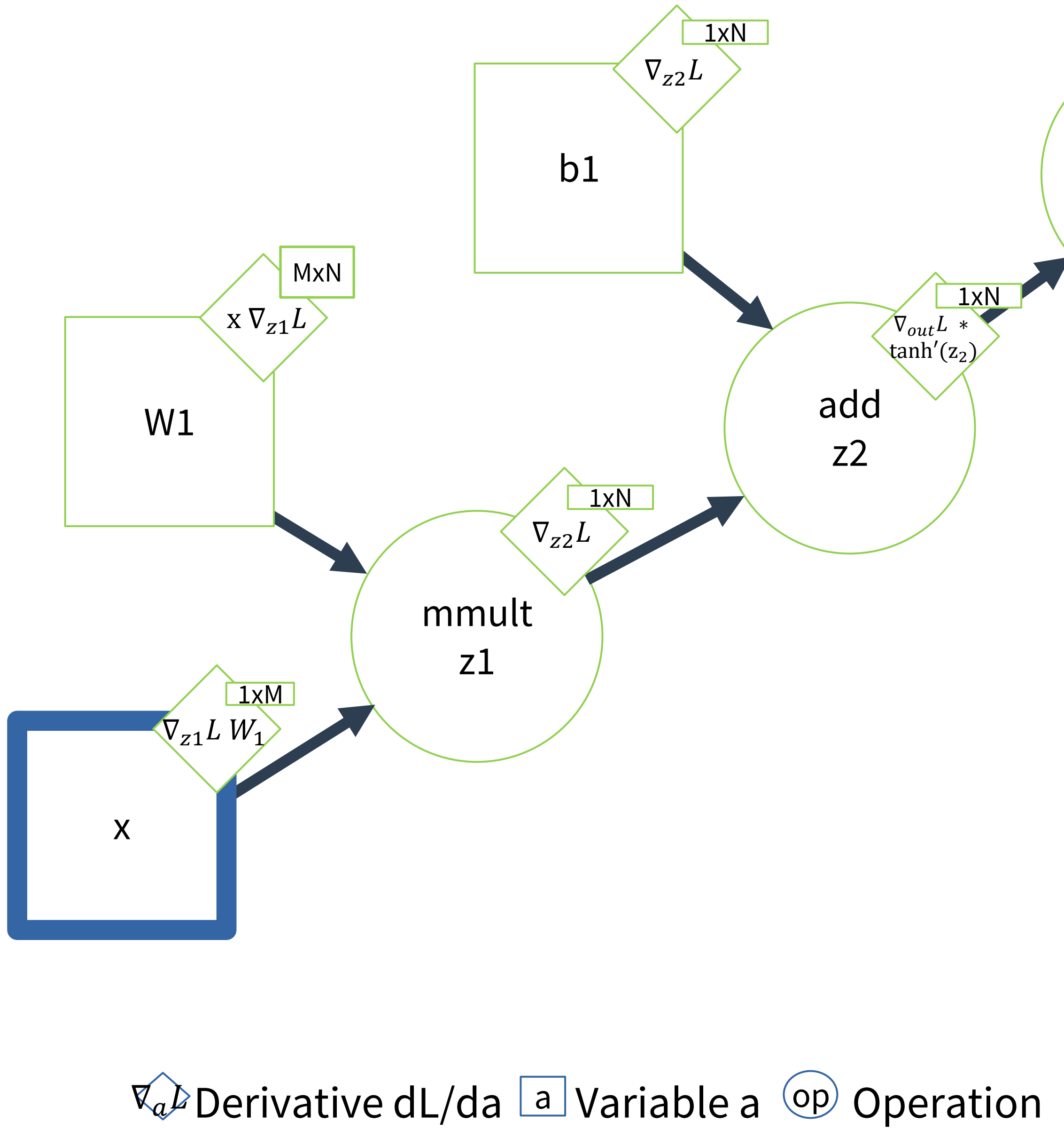


$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

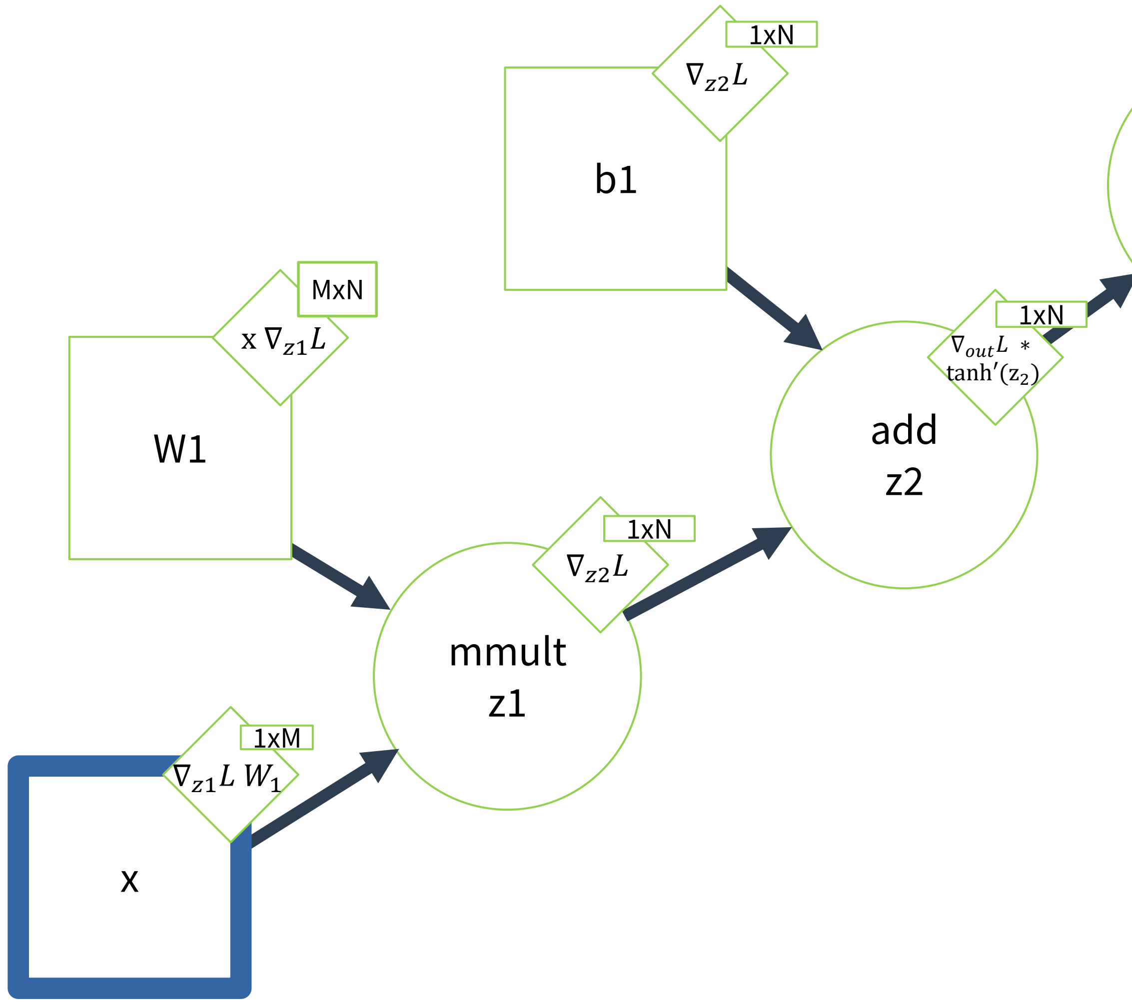
We will continue the graph search by

visiting $\boxed{x}$.



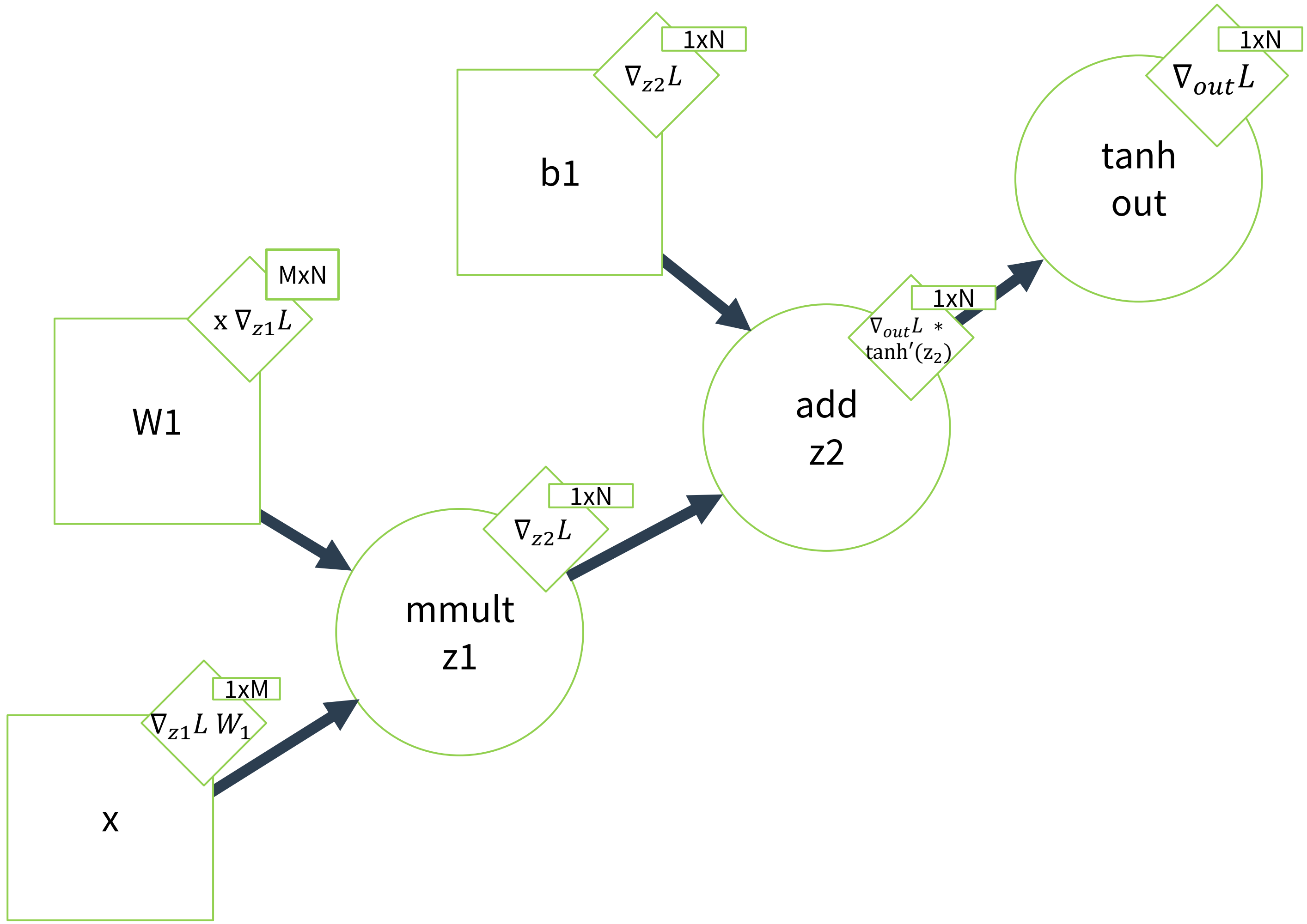
Simple MLP

$\boxed{x}$ has no gradient-enabled parents, and we don't care about its gradient, so we do nothing.



$\nabla_a L$ Derivative dL/da   a Variable a   op Operation

Simple MLP



$\nabla_a L$ Derivative dL/da a Variable a op Operation

Simple MLP

$\nabla_{out}L$ 1xN from loss function

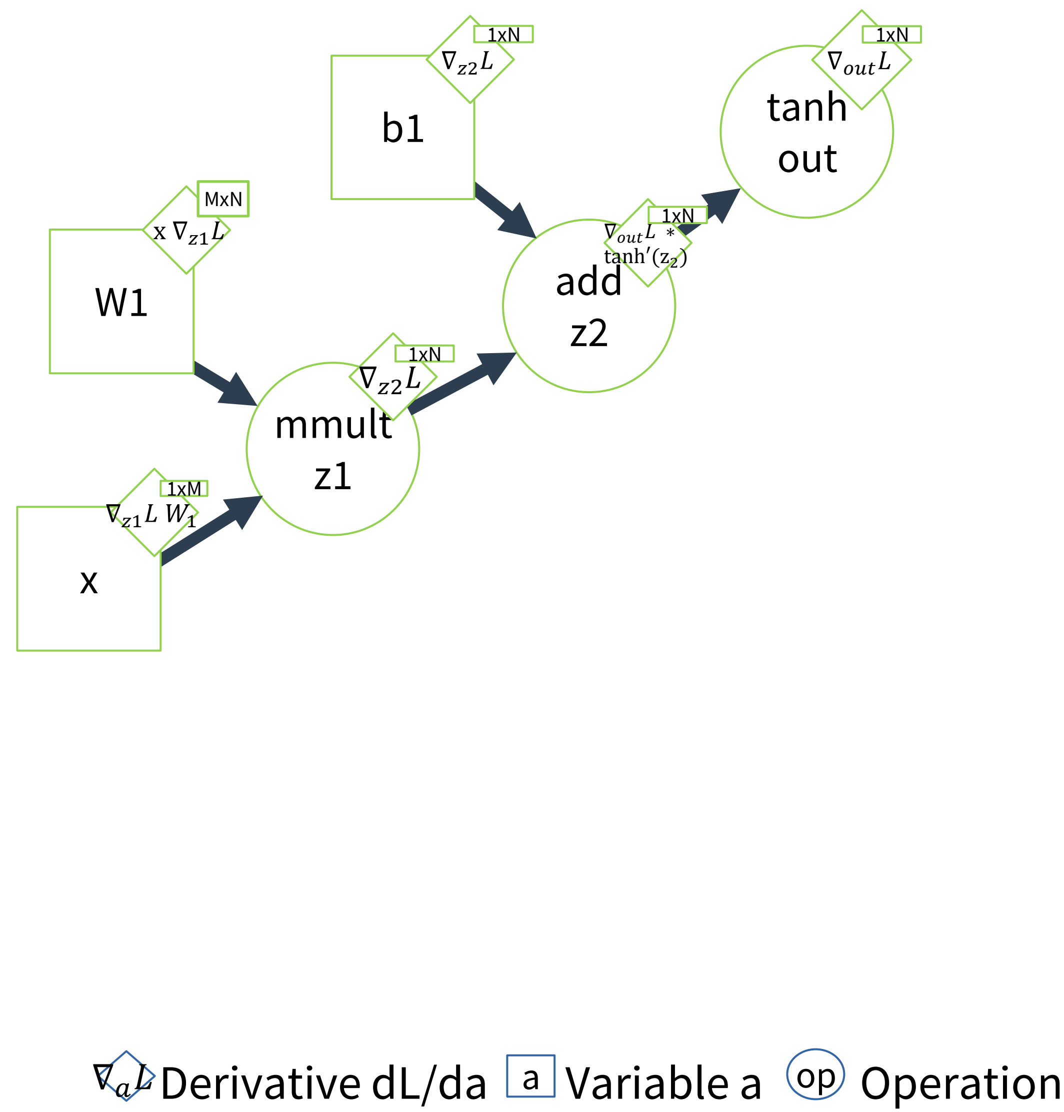
$\nabla_{z2}L$ 1xN $= \nabla_{z3}L\; tanh'(z_2)$

$\nabla_{b1}L$ 1xN $= \nabla_{z2}L$

$\nabla_{z1}L$ 1xN $= \nabla_{z2}L$

$\nabla_{W1}L$ MxN $= x\nabla_{z1}L$

$\nabla_{x}L$ 1xM $= \nabla_{z1}L\; W_1$



Simple MLP

$\nabla_{out}L$ 1xN from loss function

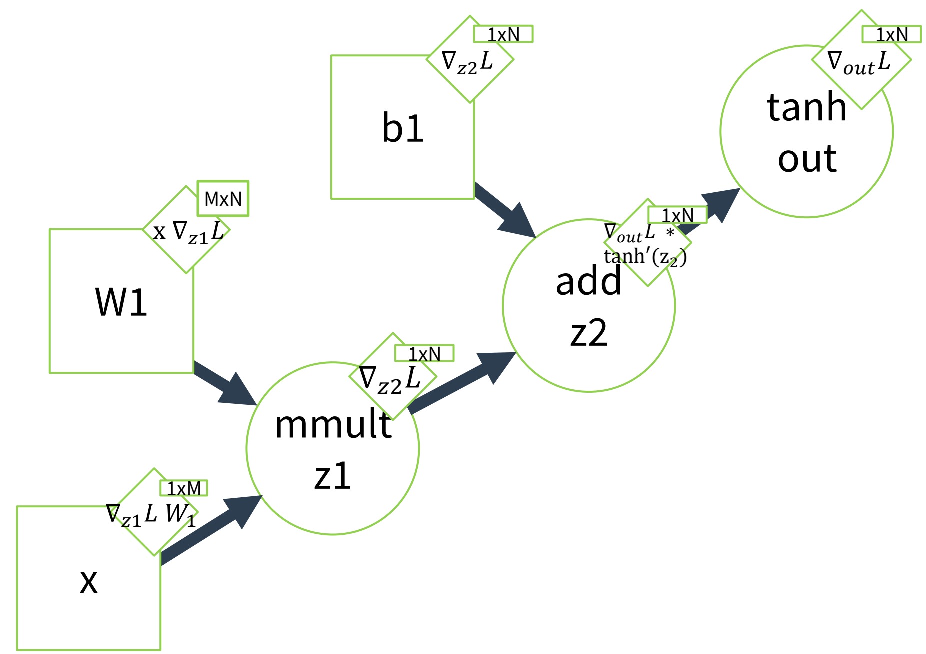
$\nabla_{z2}L$ 1xN $= \nabla_{z3}L\ tanh'(z_2)$

$\nabla_{b1}L$ 1xN $= \nabla_{z2}L$

$\nabla_{z1}L$ 1xN $= \nabla_{z2}L$

$\nabla_{W1}L$ MxN $= x\nabla_{z1}L$

$\nabla_{x}L$ 1xM $= \nabla_{z1}L\ W_1$



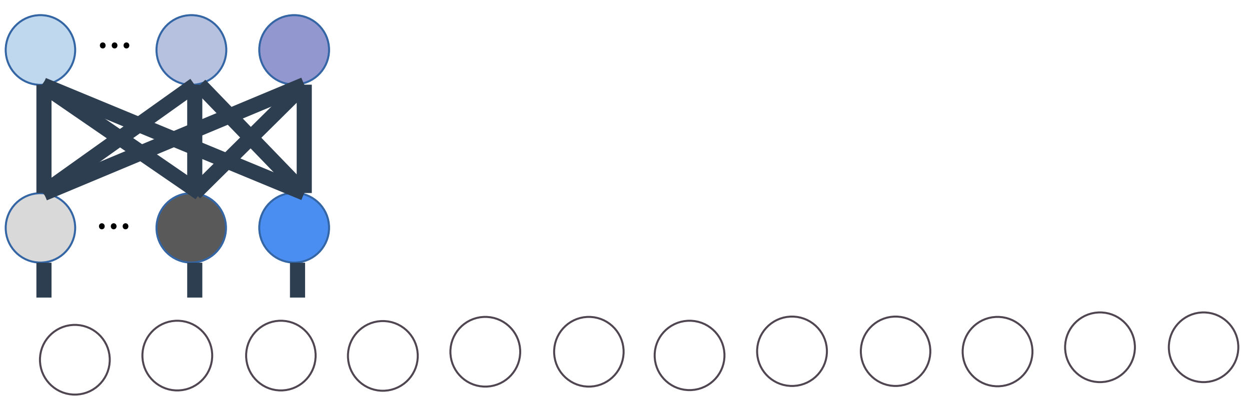
We have gradients for nodes that **accumulated (i.e. saved)** them:
W1, b1

$\nabla_a L$ Derivative dL/da a Variable a op Operation

What about reusing parameters or intermediate variables?

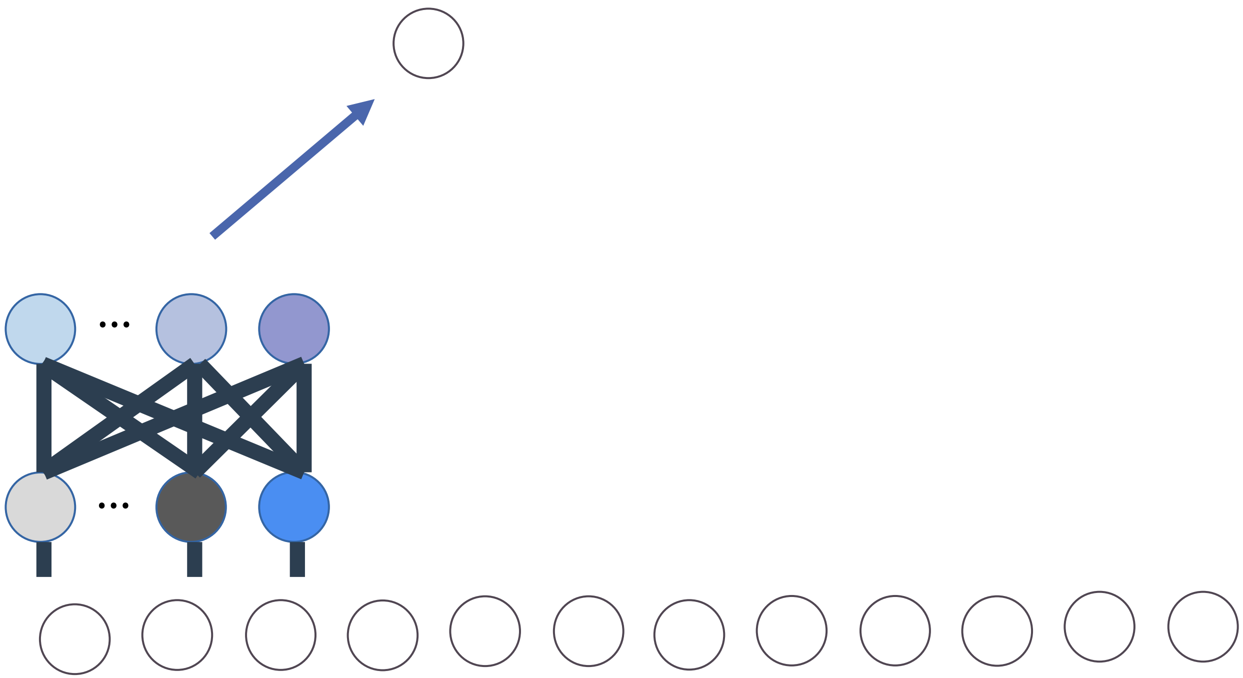
Shared Parameter Networks (Scanning MLP)

The scanning MLP "scans" across some input



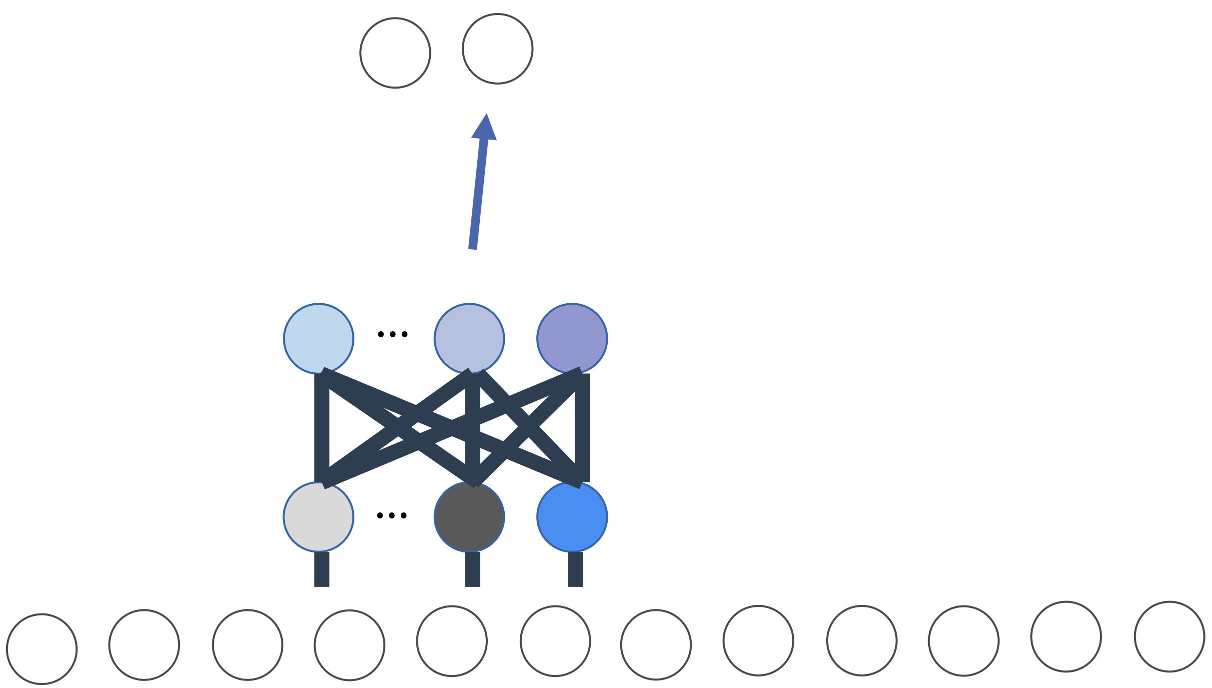
Shared Parameter Networks (Scanning MLP)

The scanning MLP "scans" across some input



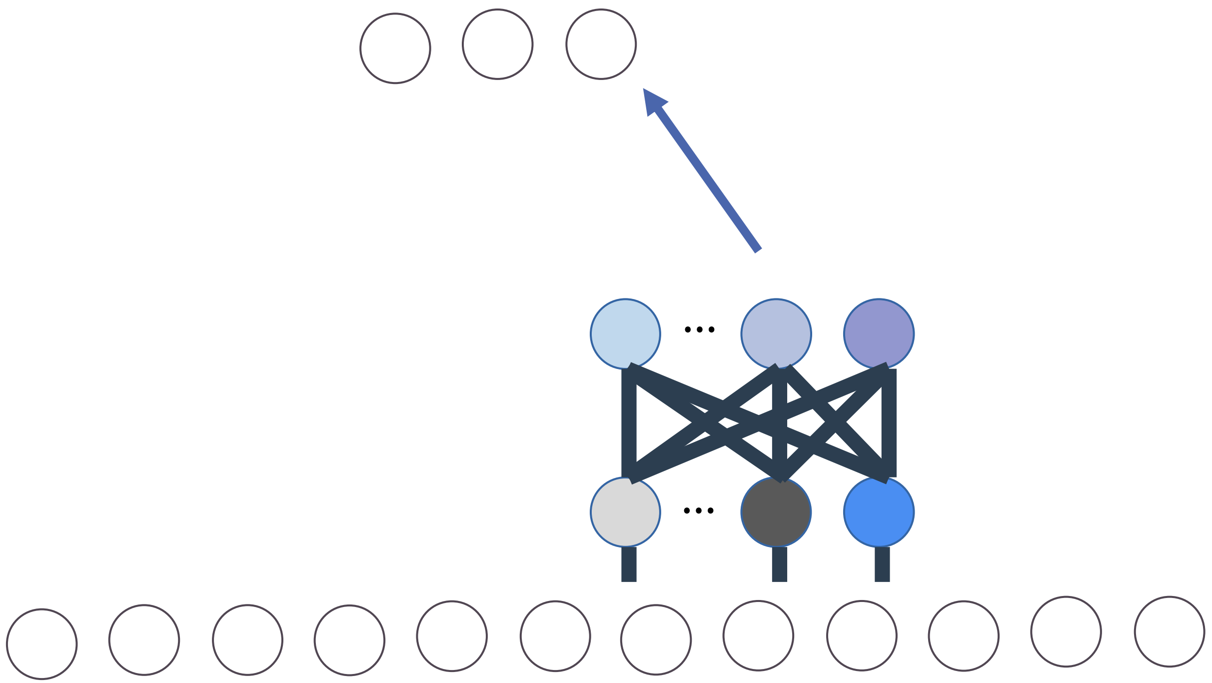
Shared Parameter Networks (Scanning MLP)

The scanning MLP “scans” across some input



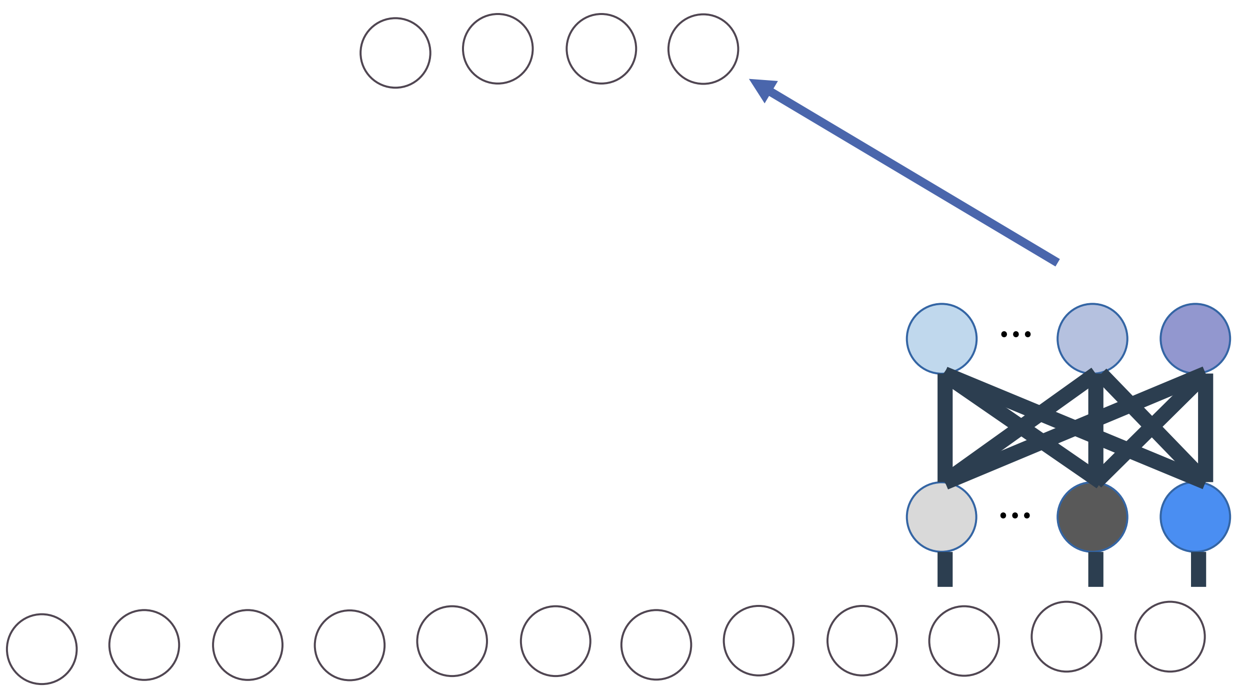
Shared Parameter Networks (Scanning MLP)

The scanning MLP "scans" across some input



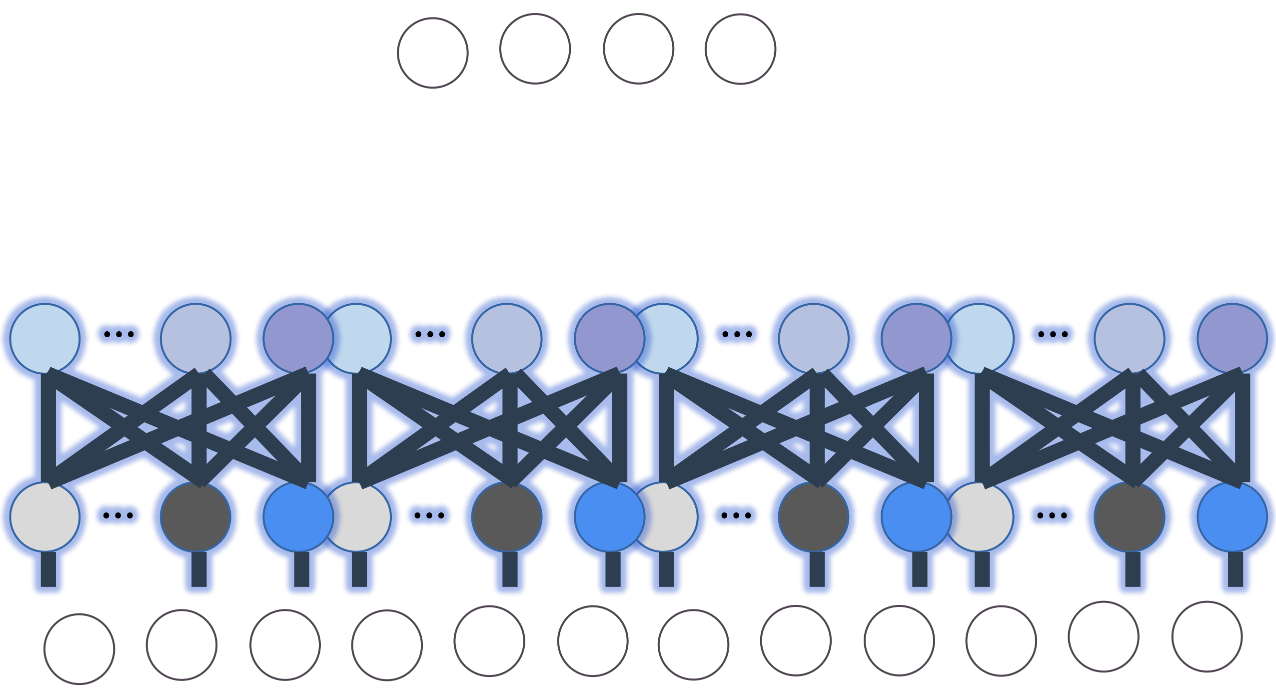
Shared Parameter Networks (Scanning MLP)

The scanning MLP "scans" across some input



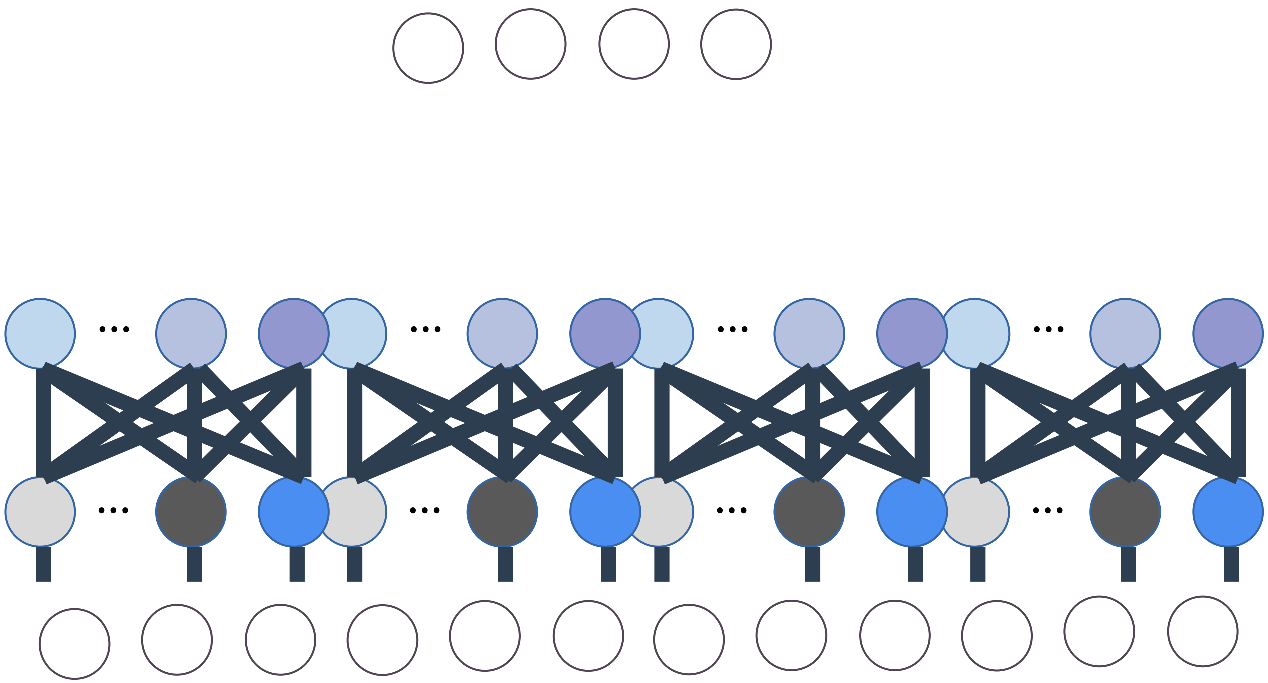
Shared Parameter Networks (Scanning MLP)

One big network with shared parameters



Shared Parameter Networks (Scanning MLP)

One big network with shared parameters



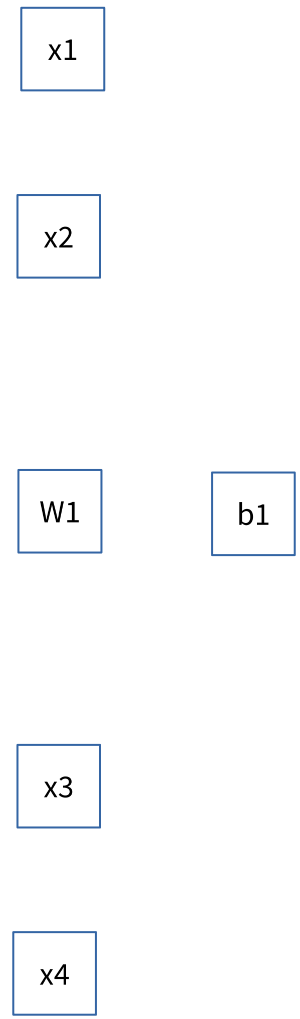
Let's create the graph...

Shared Parameter Networks (Scanning MLP)

```
for position in input:
    MLP(position)
```

Shared Parameter Networks (Scanning MLP)

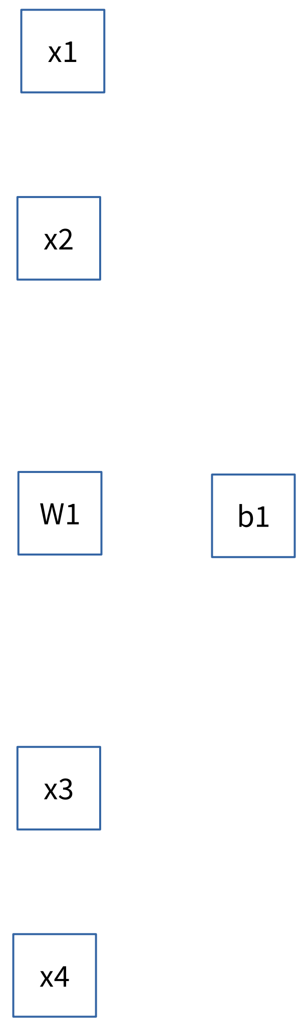
```
for position in input:
      MLP(position)
```



Our initial variables

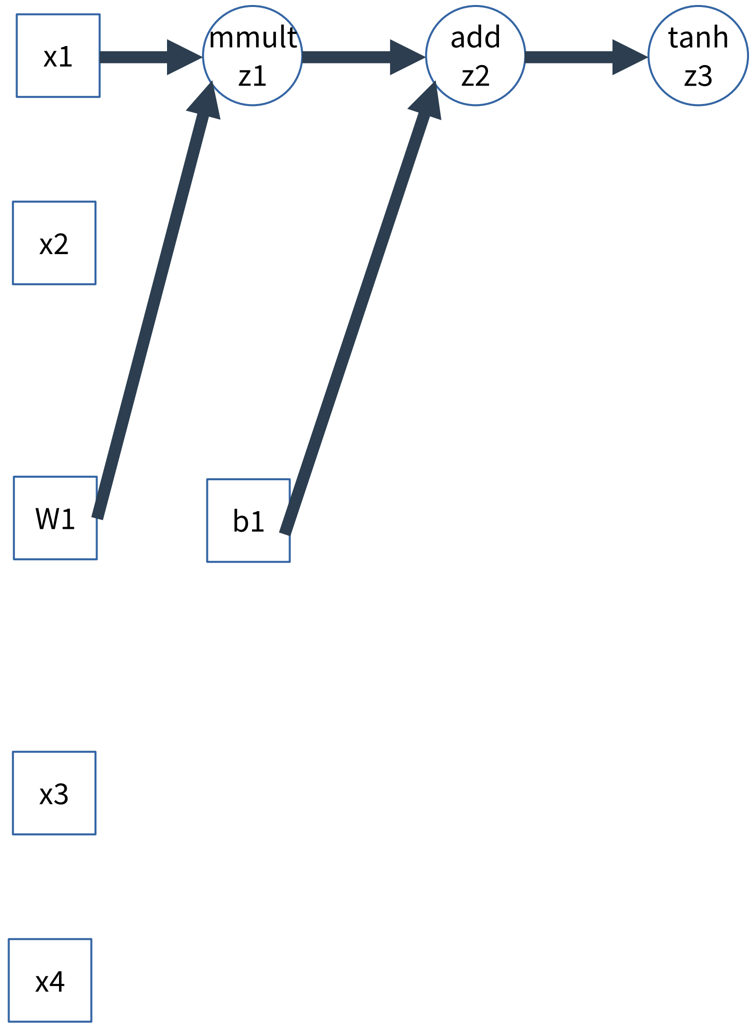
Shared Parameter Networks (Scanning MLP)

for position in input:
    MLP(position)

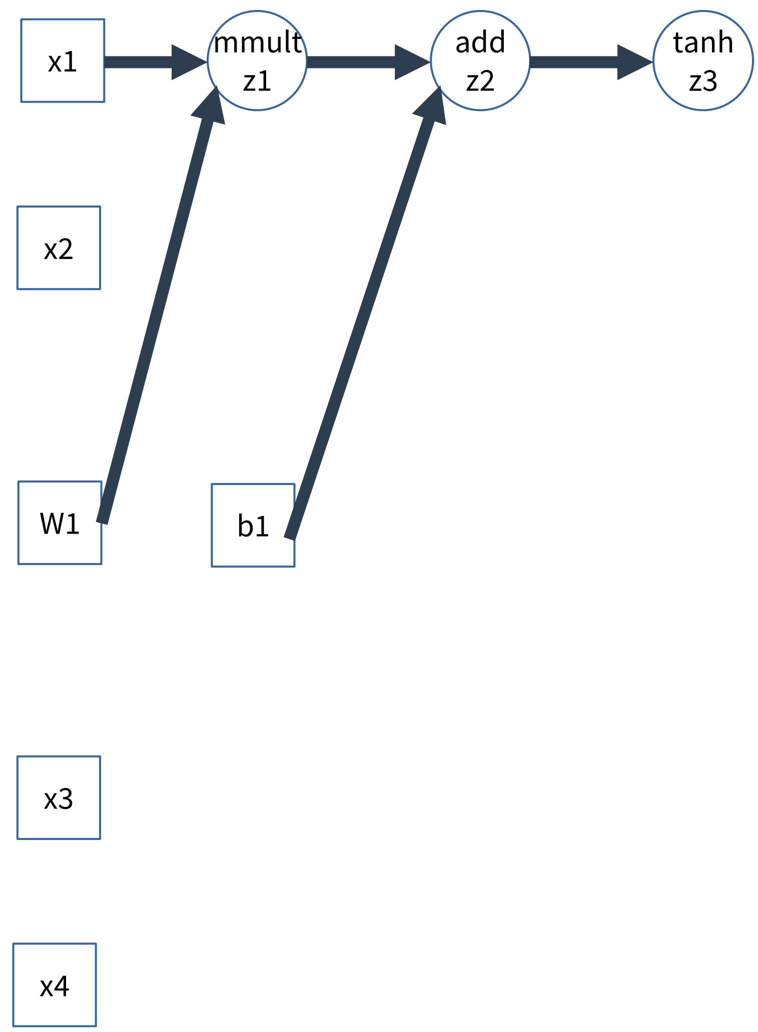


Shared Parameter Networks (Scanning MLP)

for position in input:
    MLP(position)

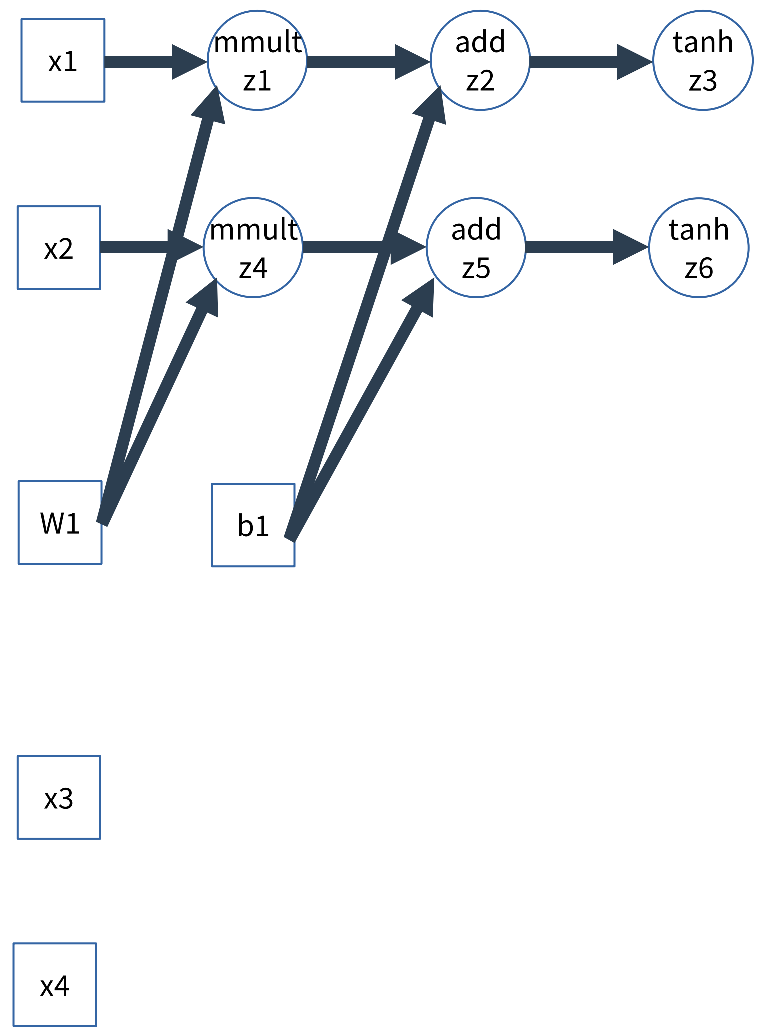


Shared Parameter Networks (Scanning MLP)



- for position in input:
  MLP(position)

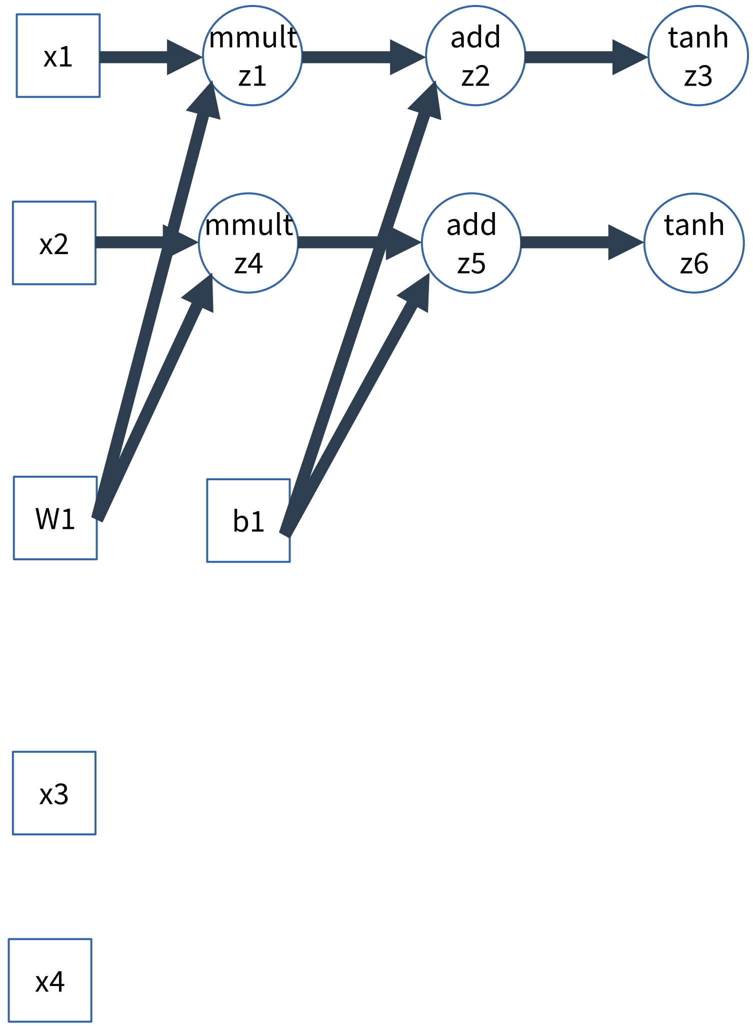
Shared Parameter Networks (Scanning MLP)



for position in input:
MLP(position)

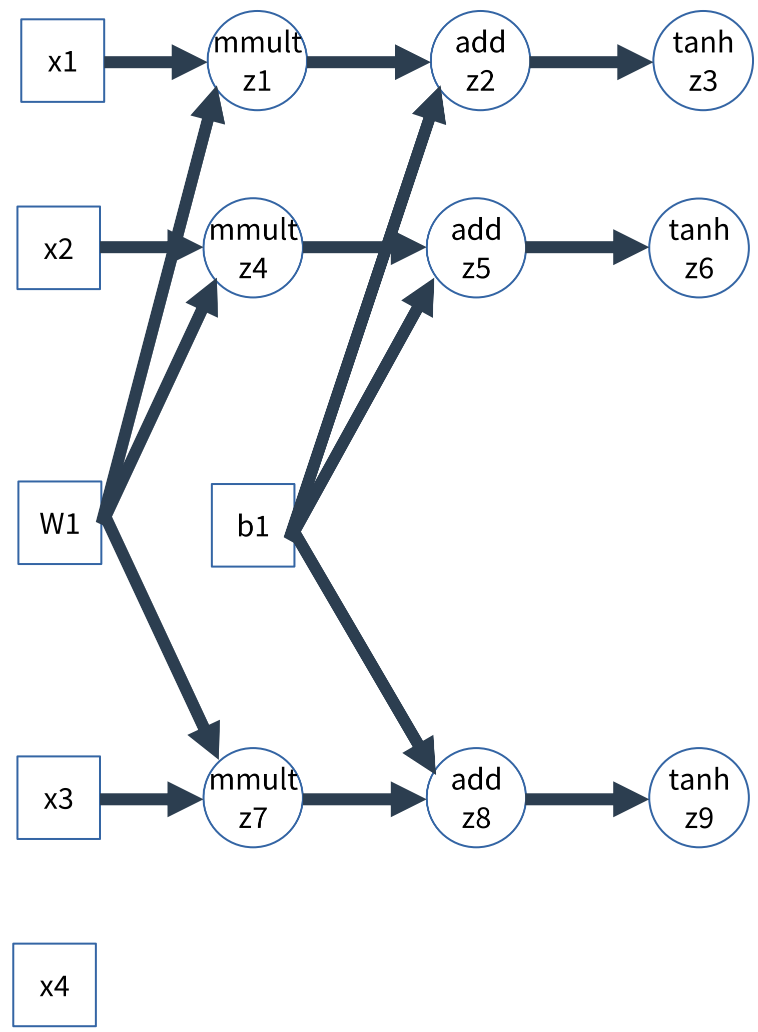
Shared Parameter Networks (Scanning MLP)

- for position in input:
  MLP(position)



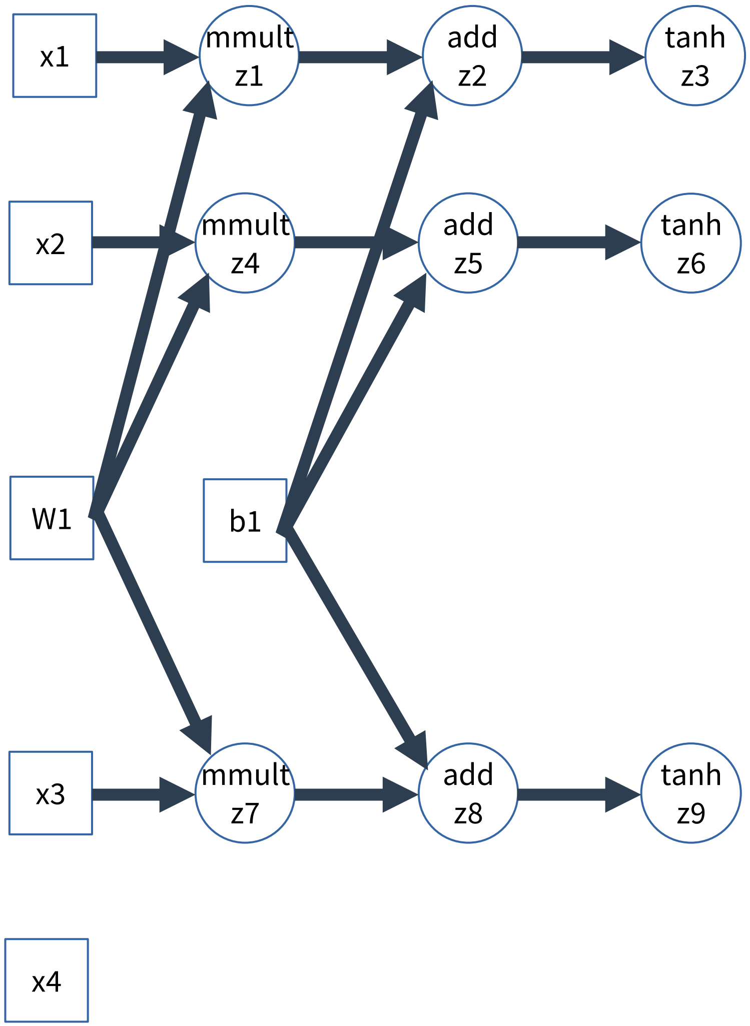
Shared Parameter Networks (Scanning MLP)

for position in input:
    MLP(position)



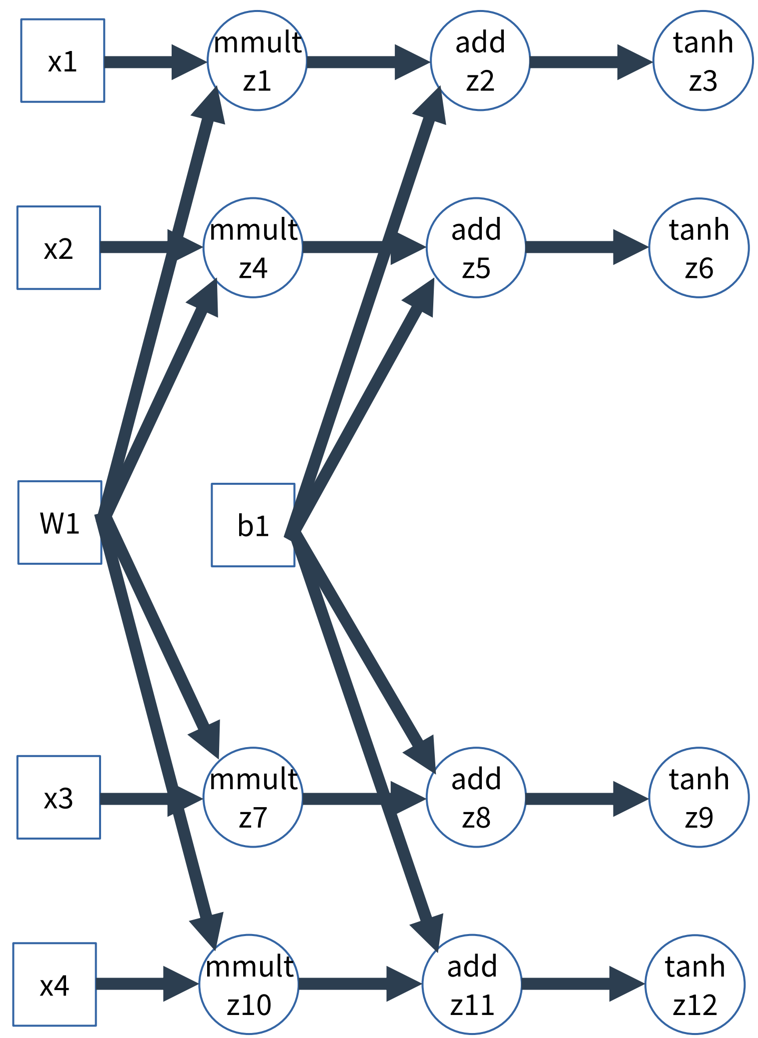
Shared Parameter Networks (Scanning MLP)

- for position in input:
  MLP(position)

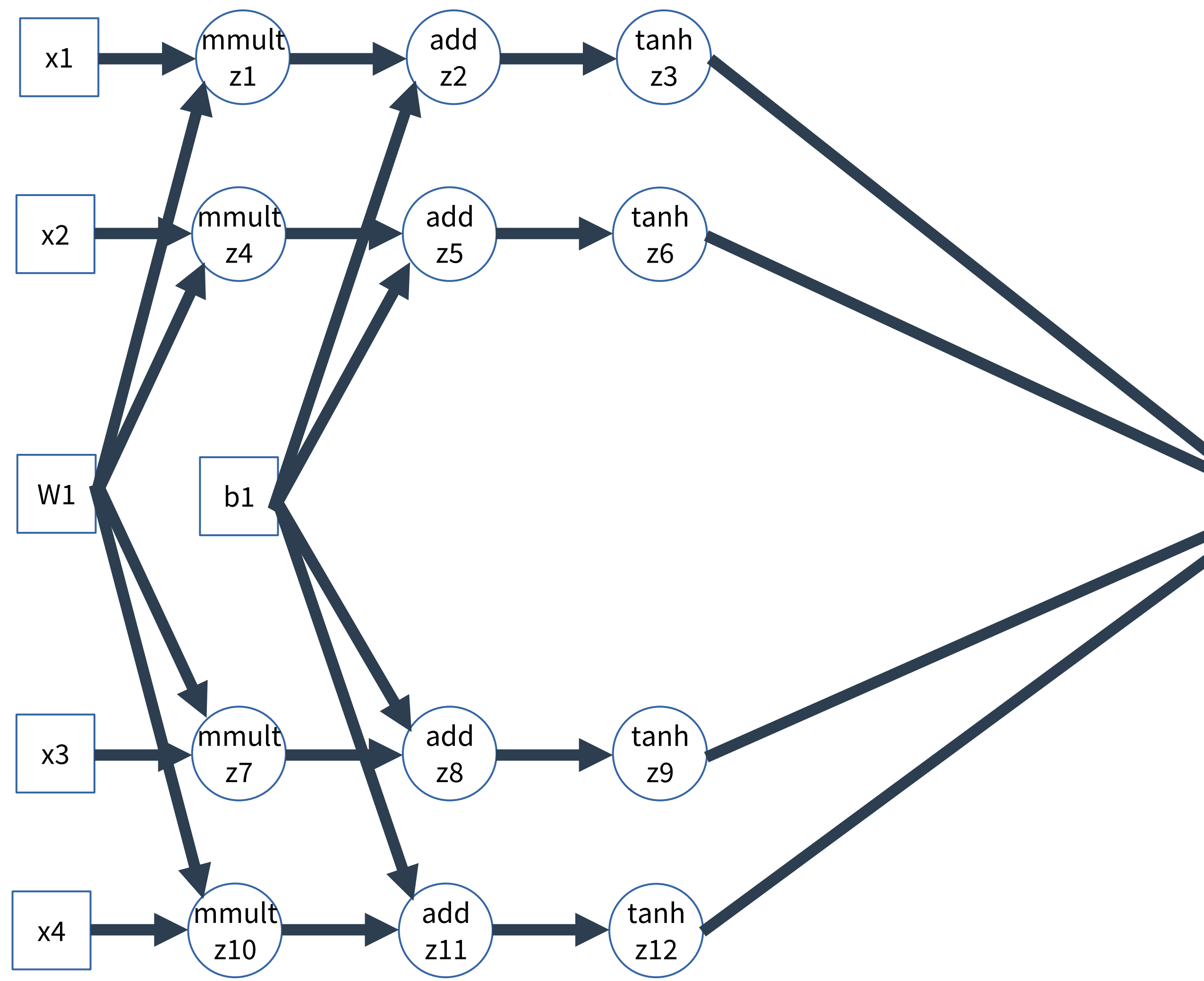


Shared Parameter Networks (Scanning MLP)

for position in input:
MLP(position)

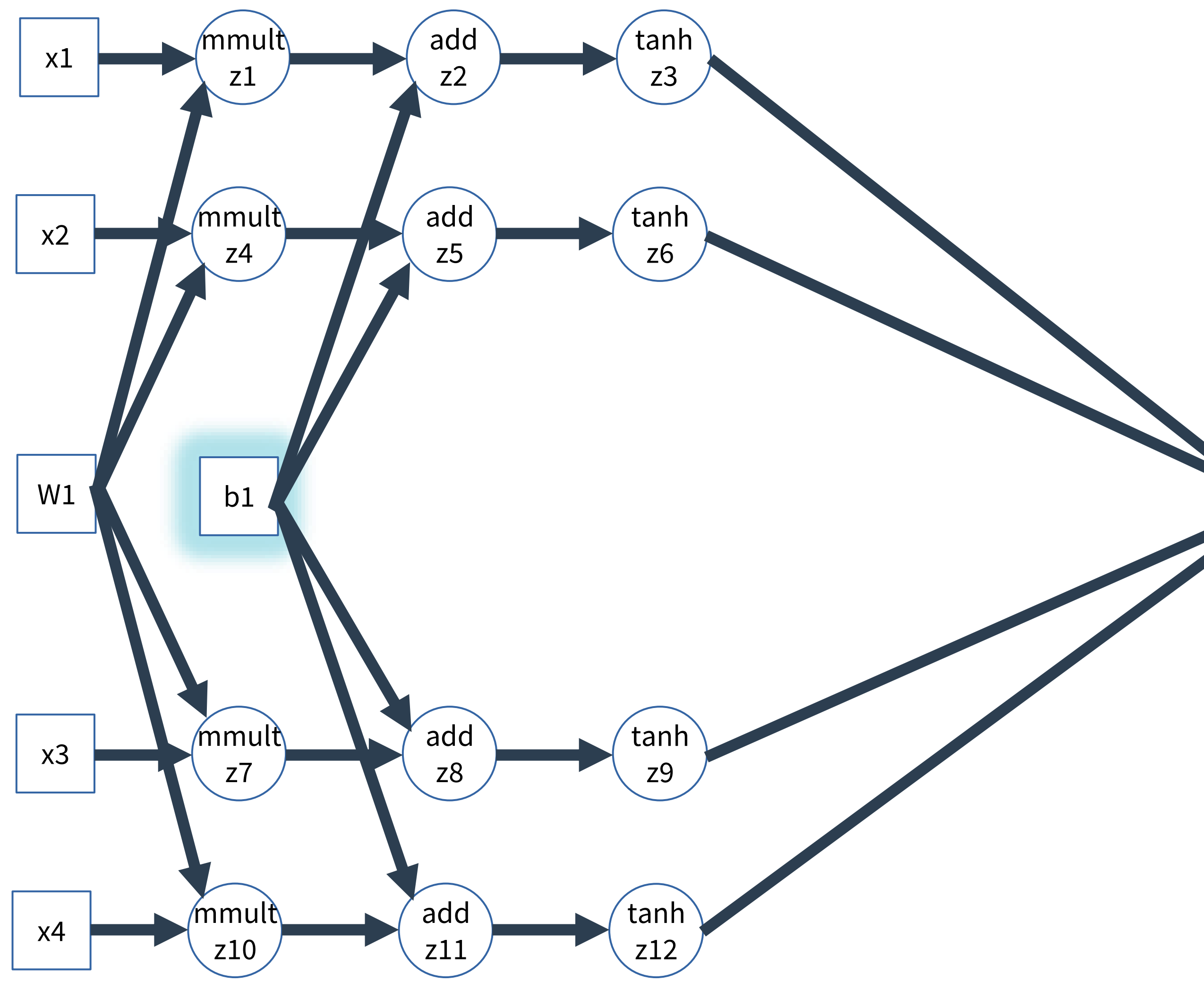


Shared Parameter Networks (Scanning MLP)



Shared Parameter Networks (Scanning MLP)

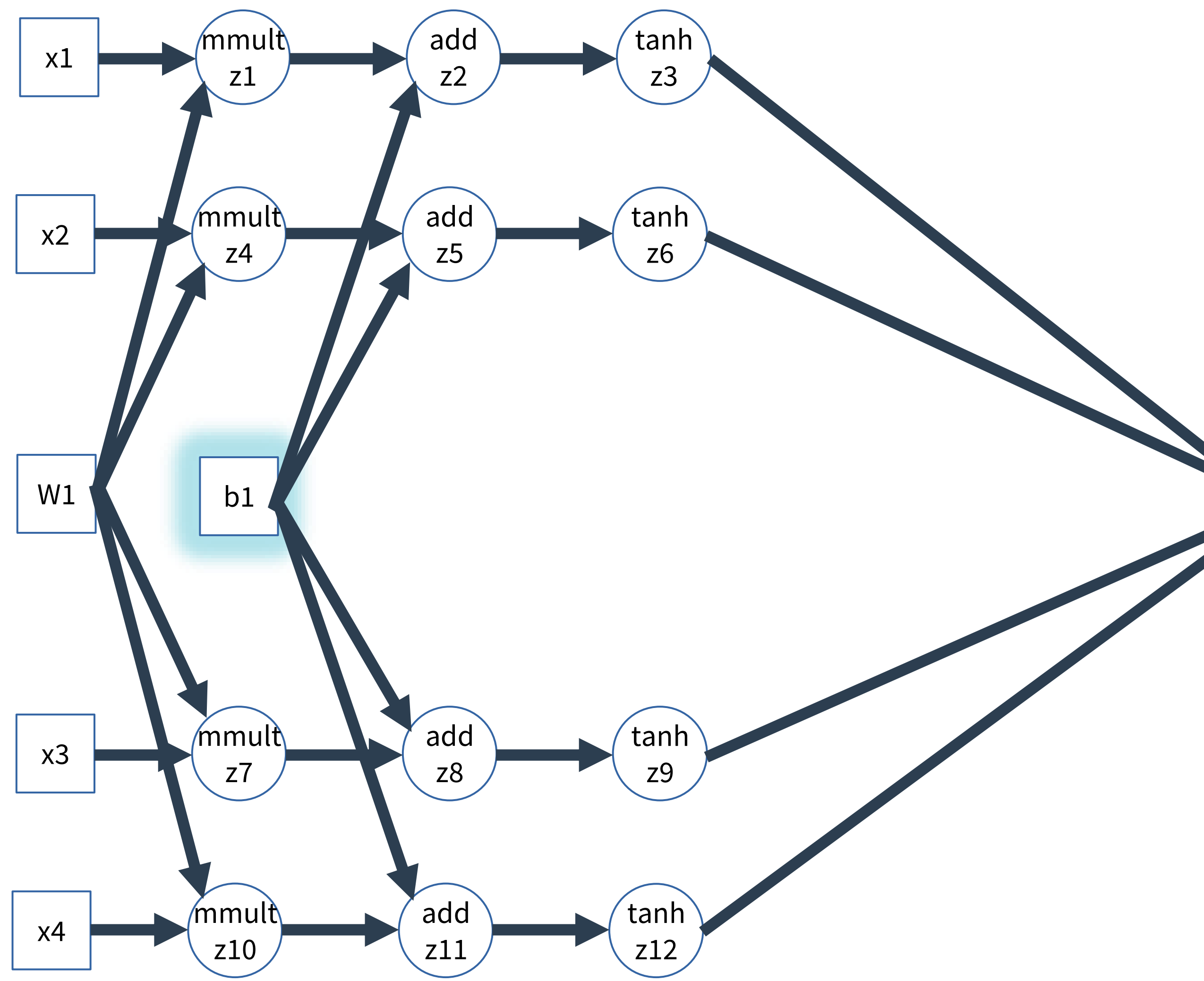
Nodes can have multiple avenues of influence



Shared Parameter Networks (Scanning MLP)

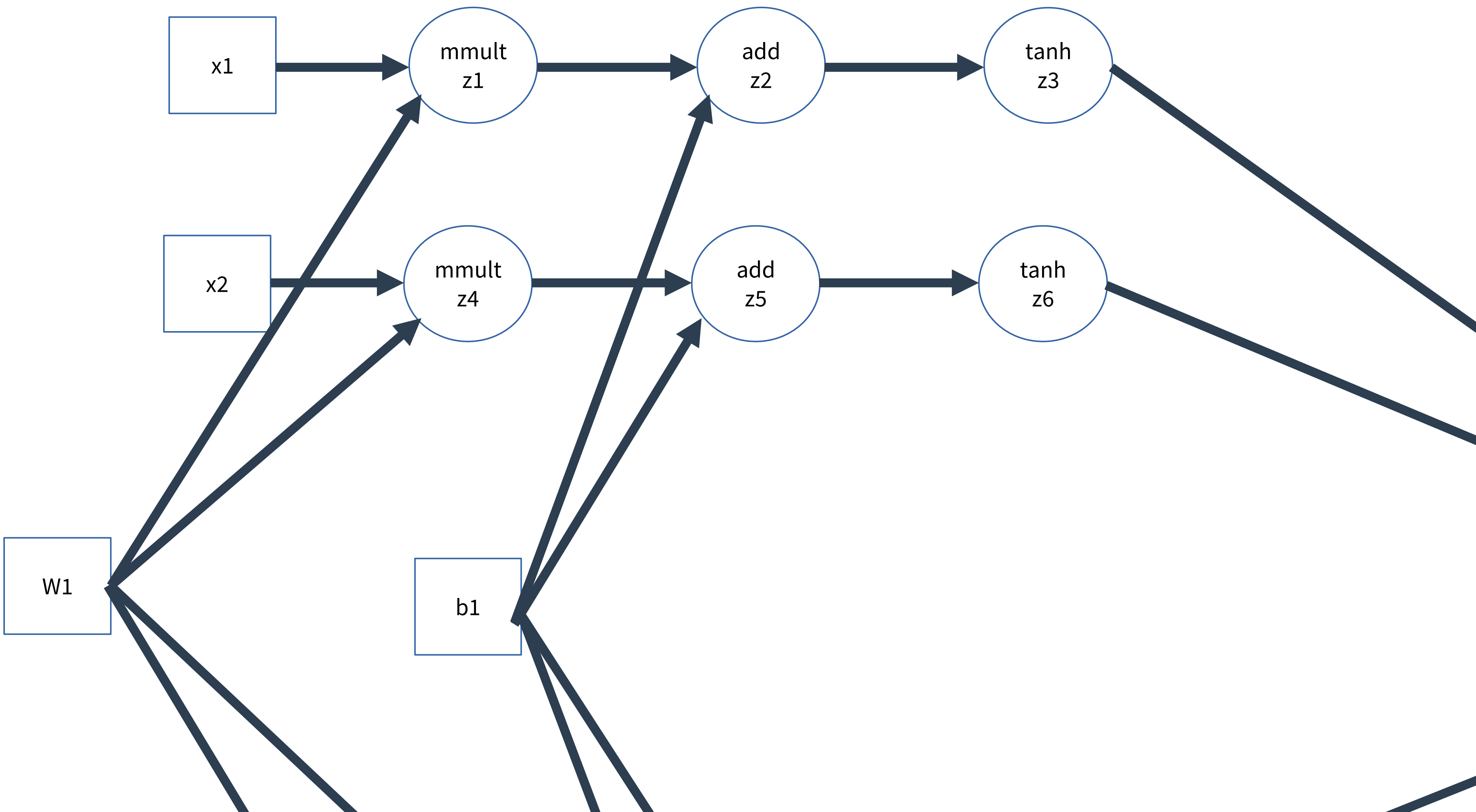
Nodes can have multiple avenues of influence

Gradient **accumulation** is especially important…



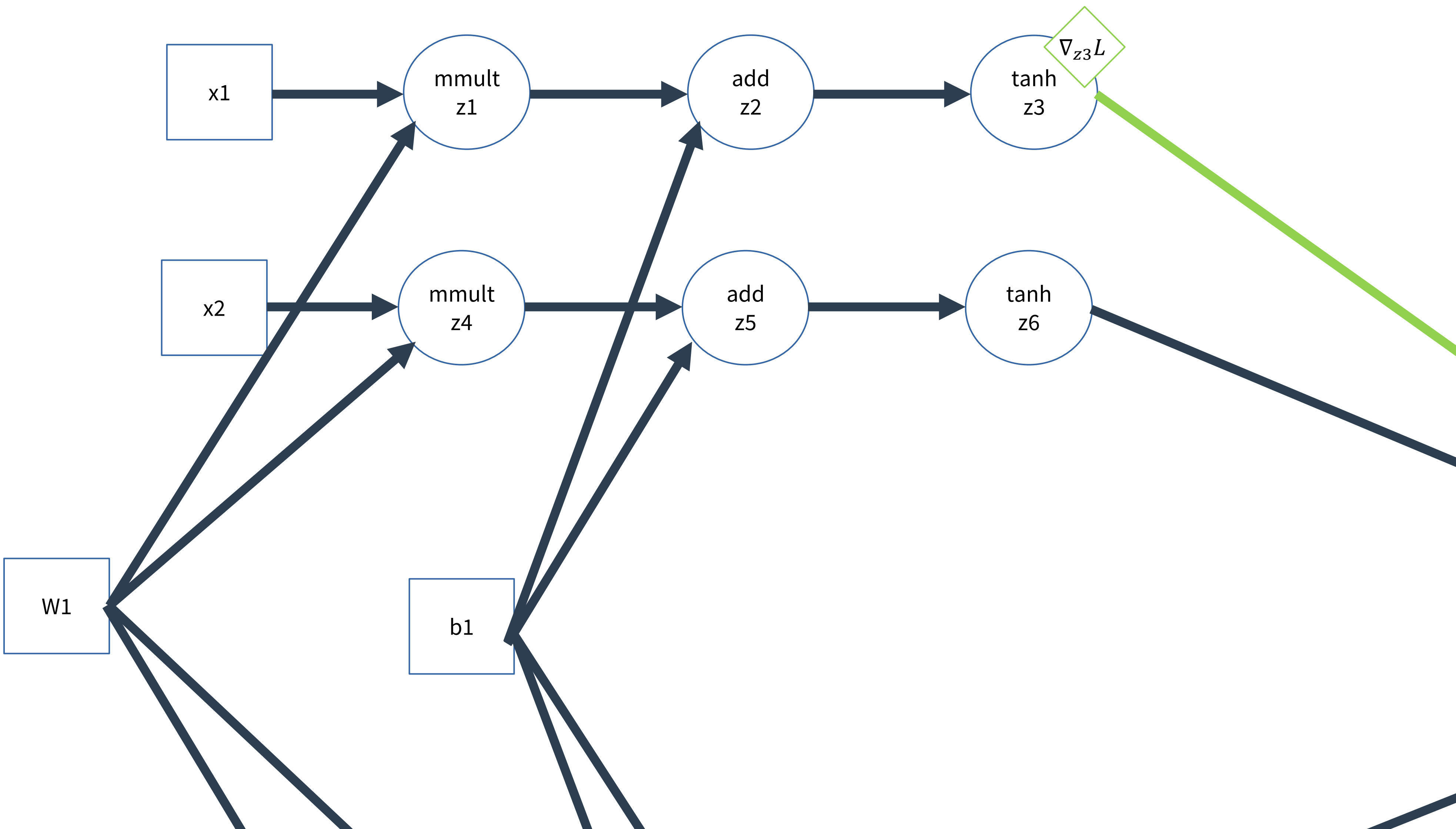
Shared Parameter Networks (Scanning MLP)

Let's DFS…



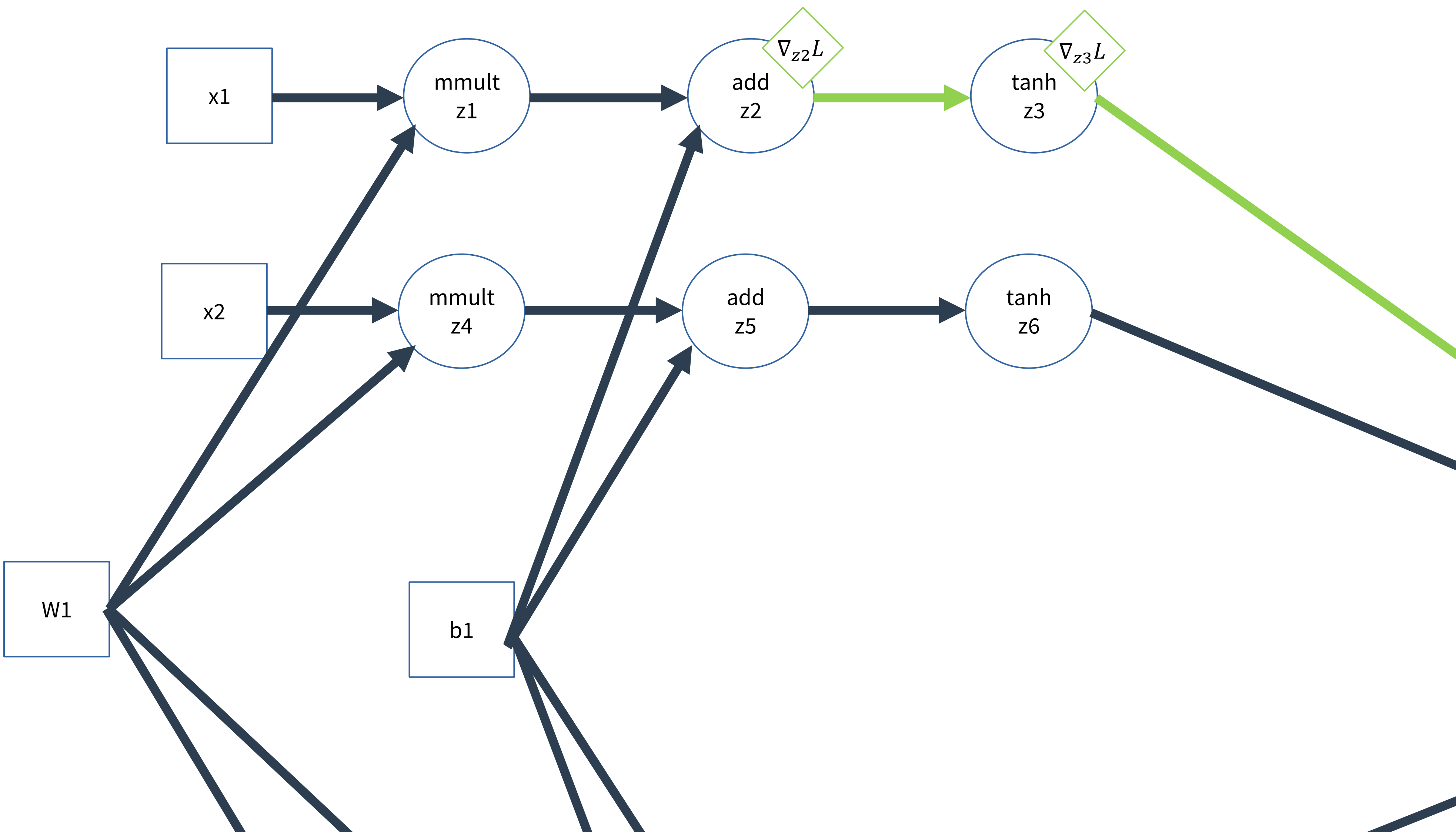
Shared Parameter Networks (Scanning MLP)

Let's DFS…



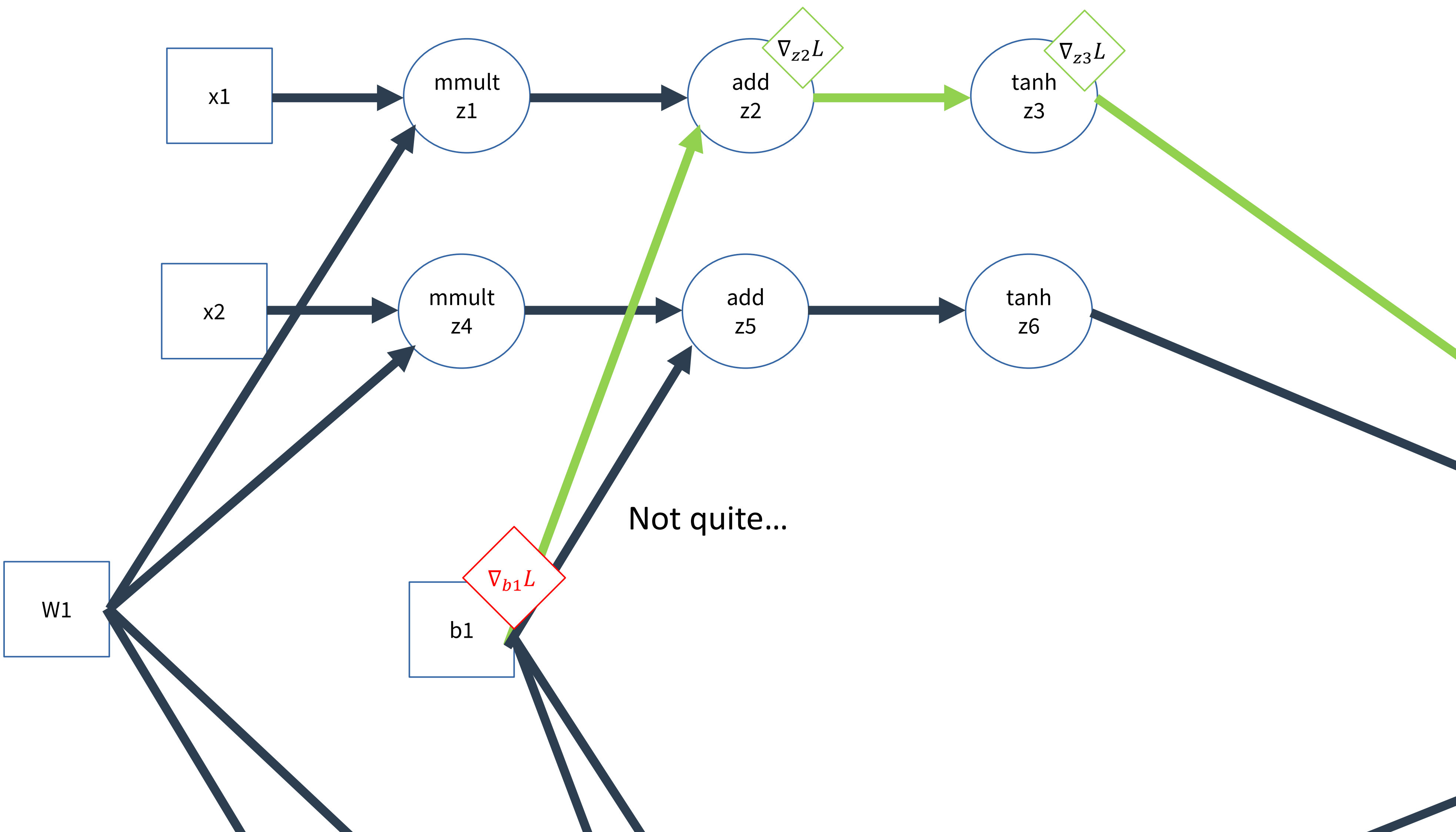
Shared Parameter Networks (Scanning MLP)

Let's DFS…



Shared Parameter Networks (Scanning MLP)

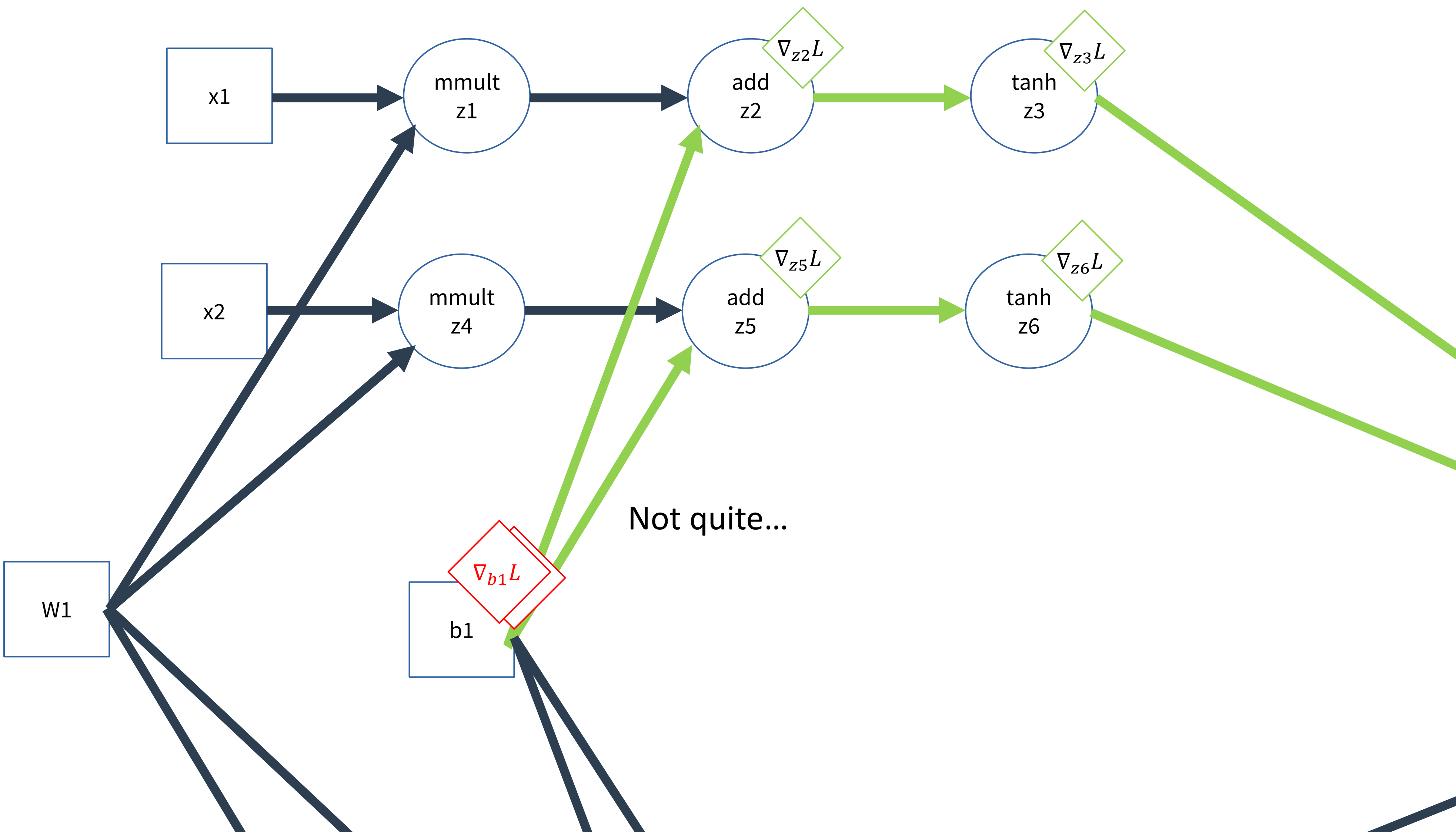
Let's DFS…



Not quite…

Shared Parameter Networks (Scanning MLP)

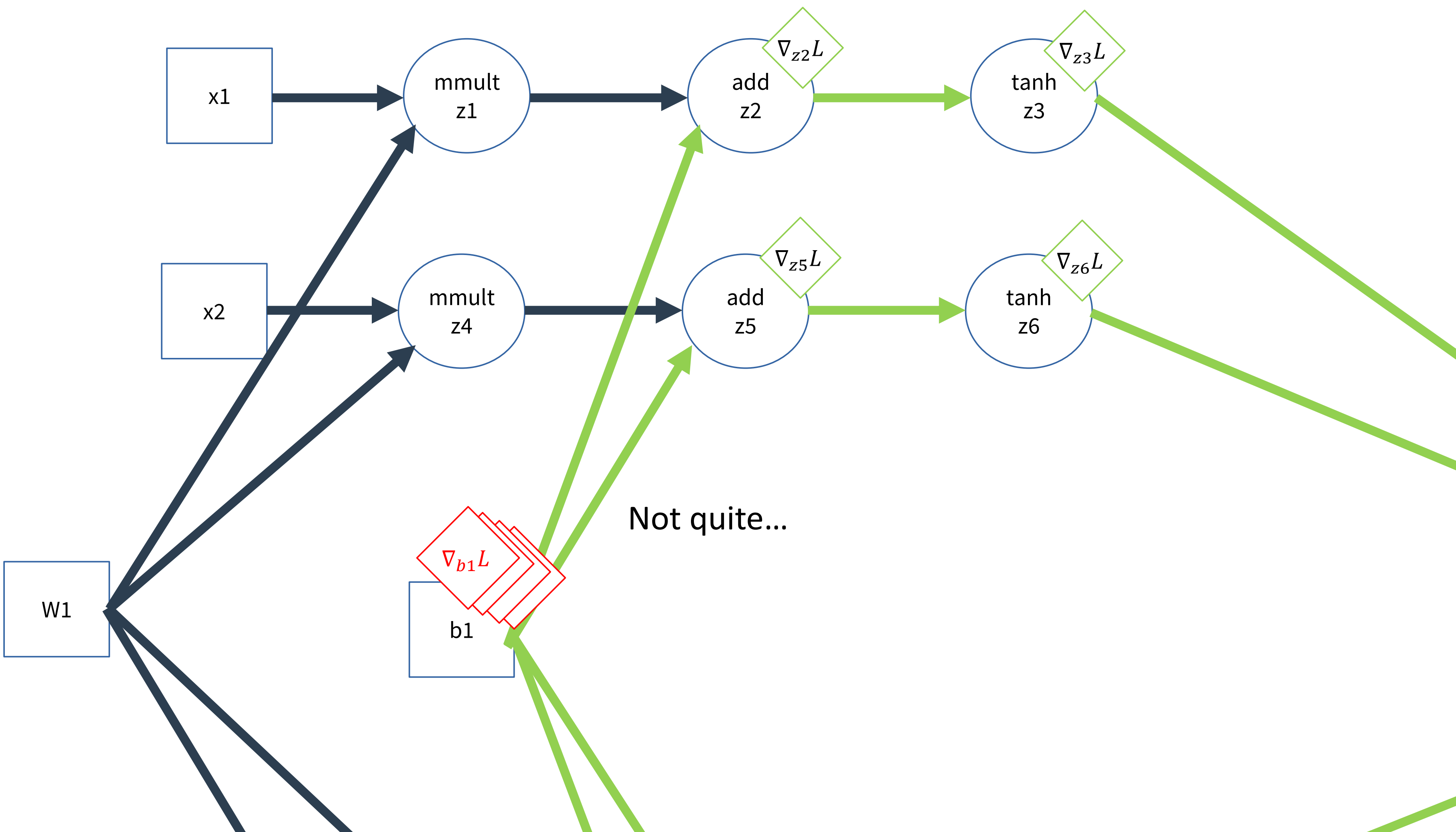
Let's DFS…



Not quite…

Shared Parameter Networks (Scanning MLP)

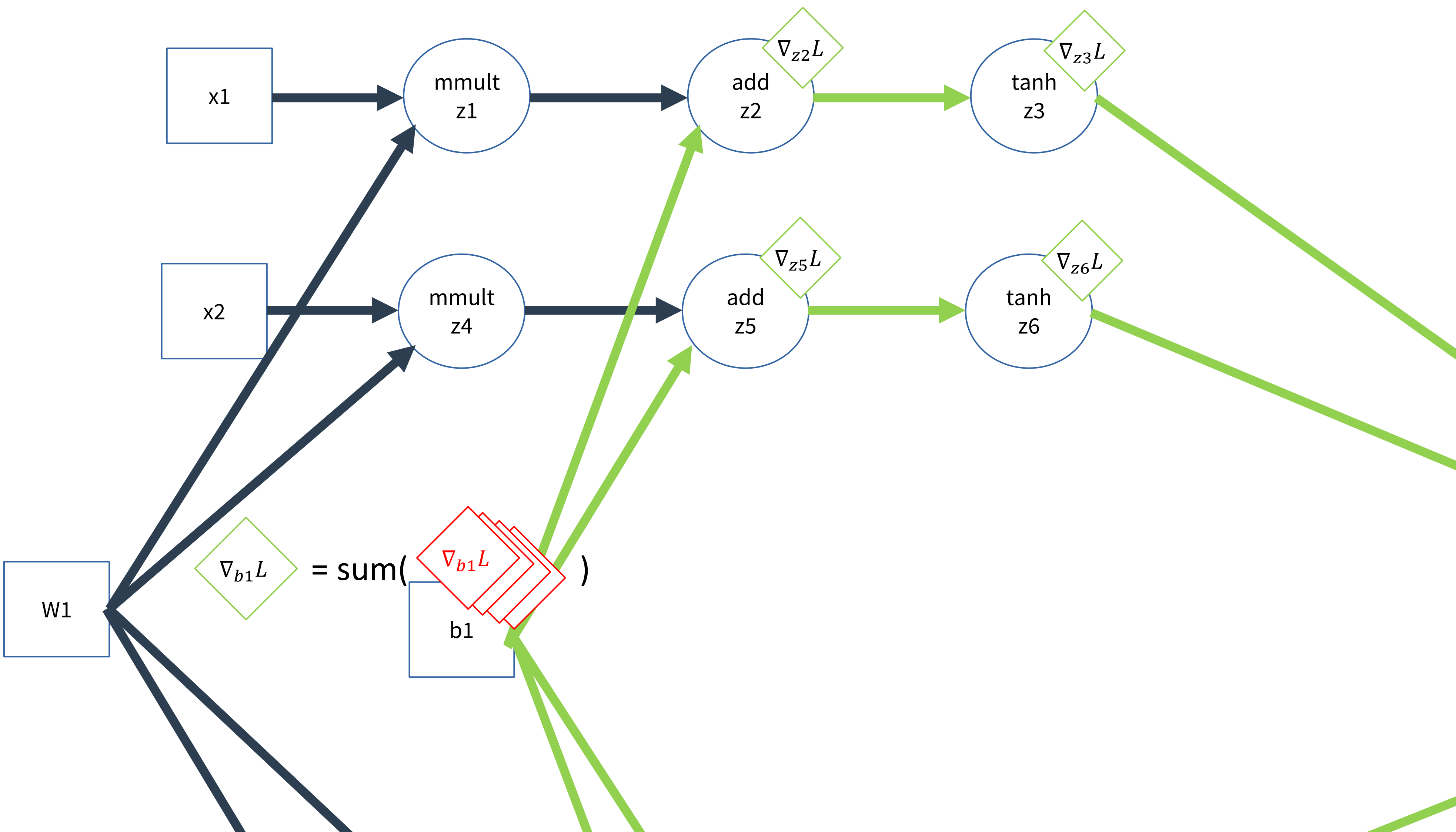
Let's DFS…



Not quite…

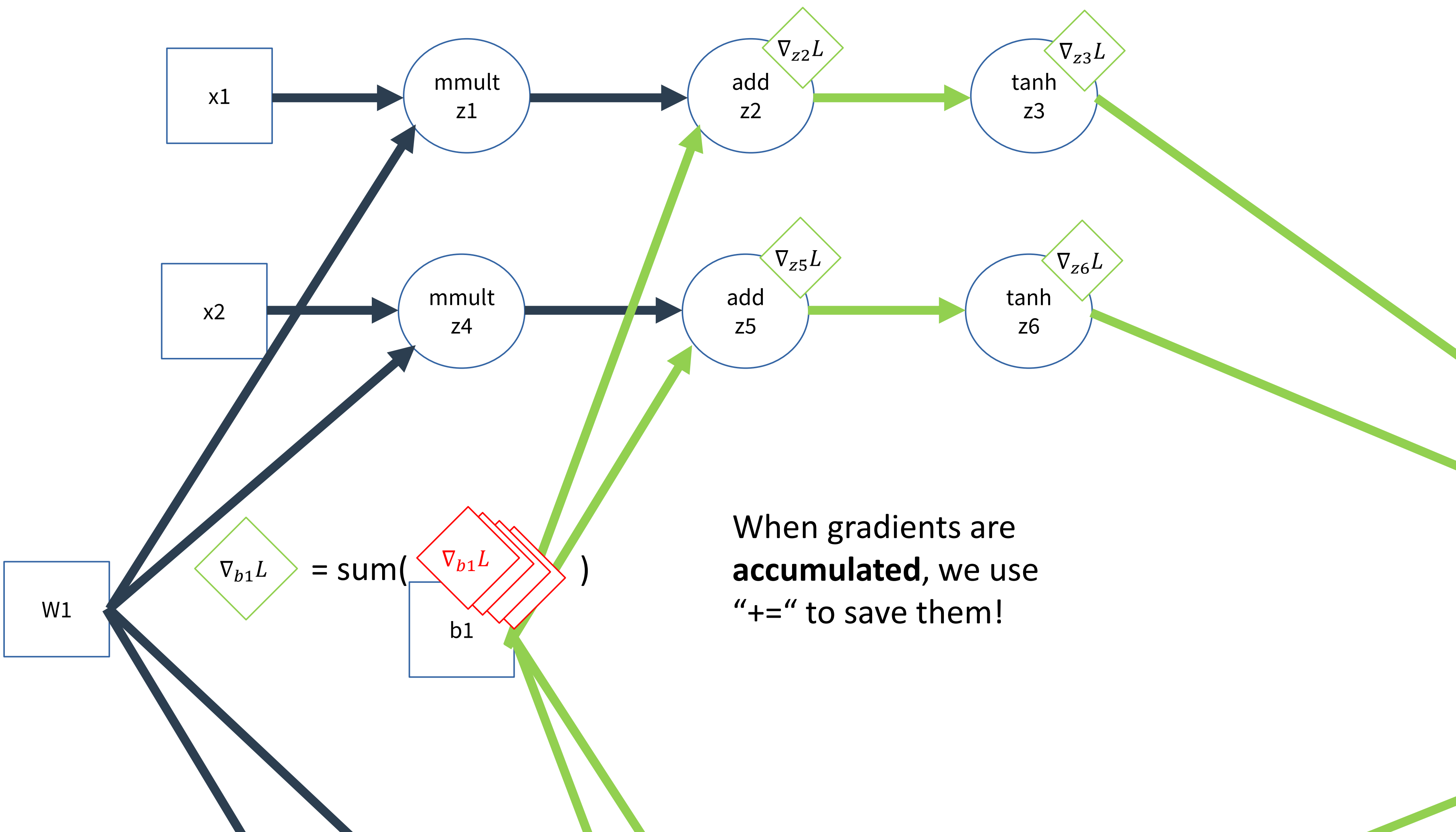
Shared Parameter Networks (Scanning MLP)

Let's DFS…



Shared Parameter Networks (Scanning MLP)

Let's DFS…



When gradients are **accumulated**, we use "+=" to save them!

Accumulating Derivatives

- Derivatives are initialized to 0 or None
- When we visit a node, we always use “+=“ to update the derivative

Accumulating Derivatives

- Derivatives are initialized to 0 or None
- When we visit a node, we always use “+=“ to update the derivative

The rest of the scanning MLP example is nothing new

We can apply this process to any function made up of smaller differentiable functions

What is this called?

- We create a graph of operations
- We graph search from known gradients
- We accumulate gradients
- We utilize reusable, differentiable operations

What is this called?

- We create a graph of operations
- We graph search from known gradients
- We accumulate gradients
- We utilize reusable, differentiable operations

Autograd

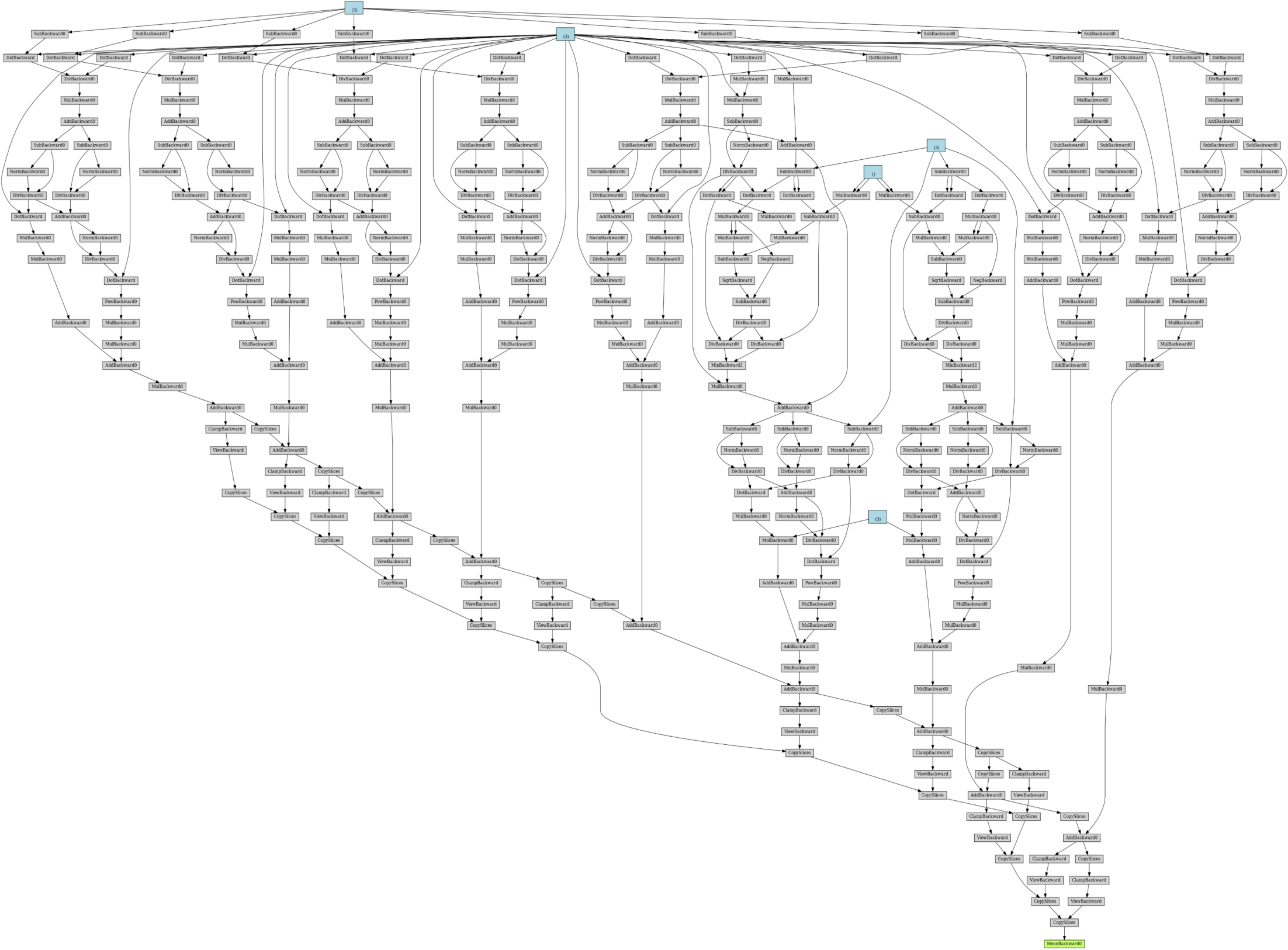
Autograd

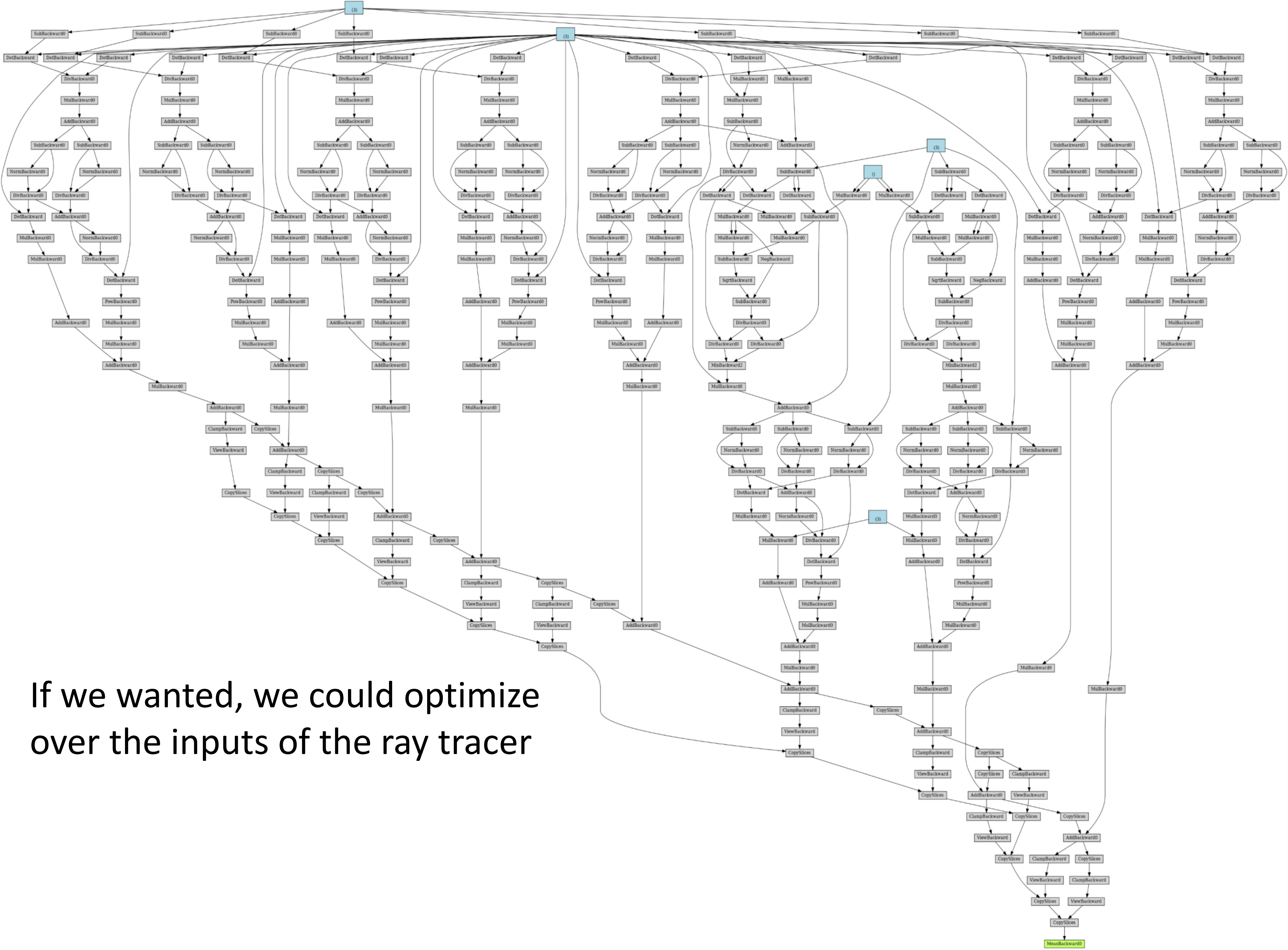
- Pytorch builds an implicit graph when you perform operations (also hw1p1)
  - +, -, *, /
  - Batchnorm, Softmax…
- You can also build this graph on paper to calculate derivatives

As an example, we'll show the graph for a ray tracer for 4x3 images

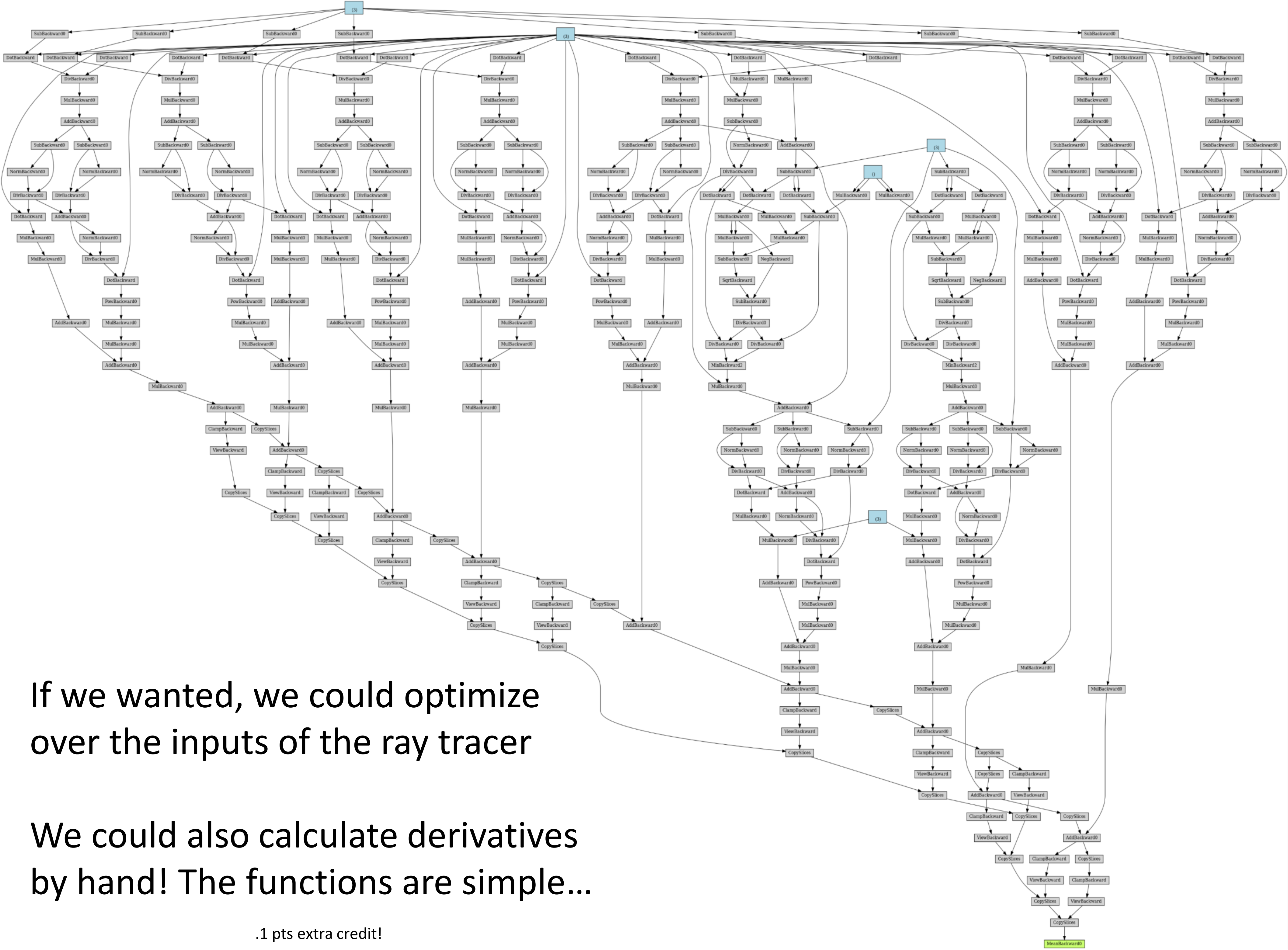
As an example, we’ll show the graph for a ray tracer for 4x3 images

Note that it has no learnable parameters





If we wanted, we could optimize over the inputs of the ray tracer



If we wanted, we could optimize over the inputs of the ray tracer

We could also calculate derivatives by hand! The functions are simple…

.1 pts extra credit!