University of BRISTOL

BACKPROPAGATION

Gabriella Miles

bristol.ac.uk

Ask questions: menti.com
Code: 7201 0048

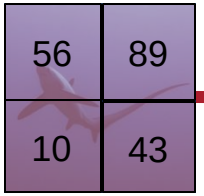
Recap

Some kind of input

Weights (or parameters)

$$f(x, W)$$

Loss → Loss Function, L



$$\begin{bmatrix} 56 & 89 \\ 10 & 43 \end{bmatrix} \rightarrow Wx \rightarrow \begin{bmatrix} 145.8 \\ -259.6 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0.2 & 0.4 & 1.3 & 2.0 \\ 0 & -2.3 & 0.1 & -1.3 \end{bmatrix} \begin{bmatrix} 56 \\ 89 \\ 10 \\ 43 \end{bmatrix} = \begin{bmatrix} 145.8 \\ -259.6 \end{bmatrix}$$

Review maths worksheet 1 if you are feeling rusty on your matrix operations!


$\nabla_w L$

Ask questions: menti.com
Code: 7201 0048

What next?

- Let's come up with a way of efficiently finding the parameters that minimise the loss function.

Backpropagation

Ask questions: menti.com
Code: 7201 0048

Objectives

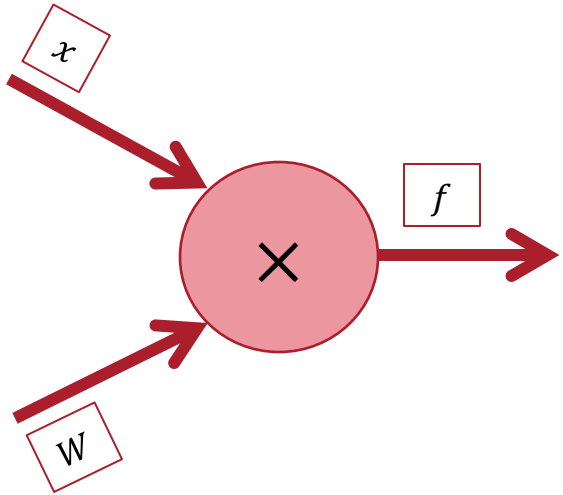
Understand:

- What backpropagation *is*
- How it *works*
- Why it's *useful* to us

Ask questions: menti.com
Code: 7201 0048

Computational Graphs

$$f(x, W) = Wx$$



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

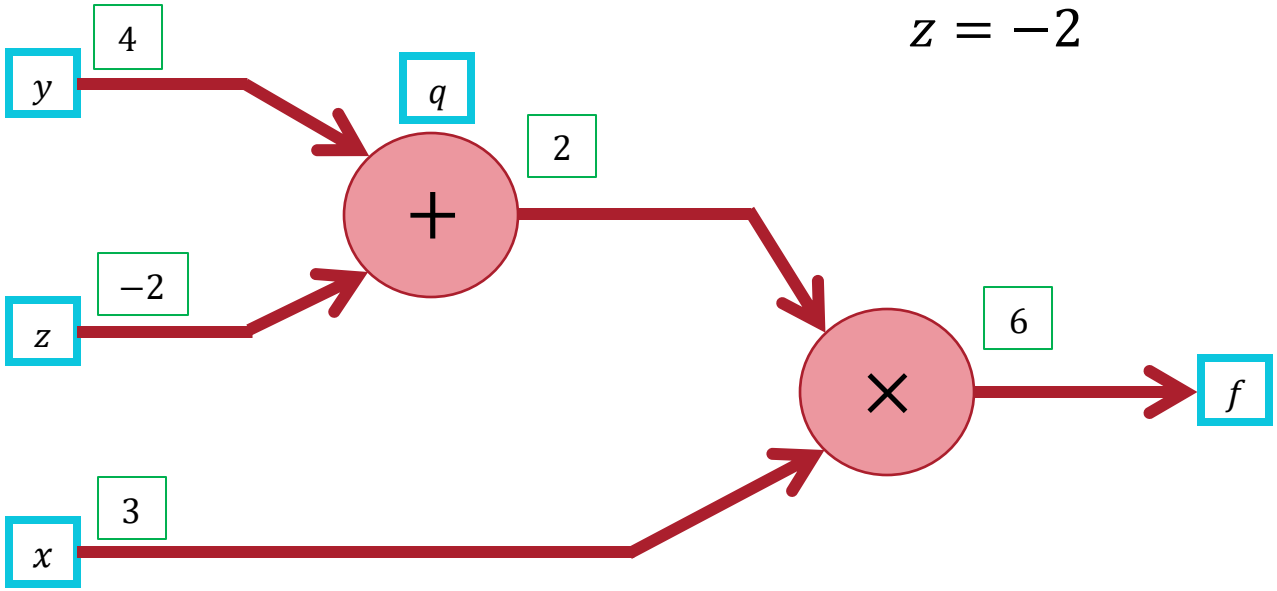
Hint: you will need 2 nodes!

Ask questions: menti.com
Code: 7201 0048

The Forward Pass

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

$$x = 3$$
$$y = 4$$
$$z = -2$$

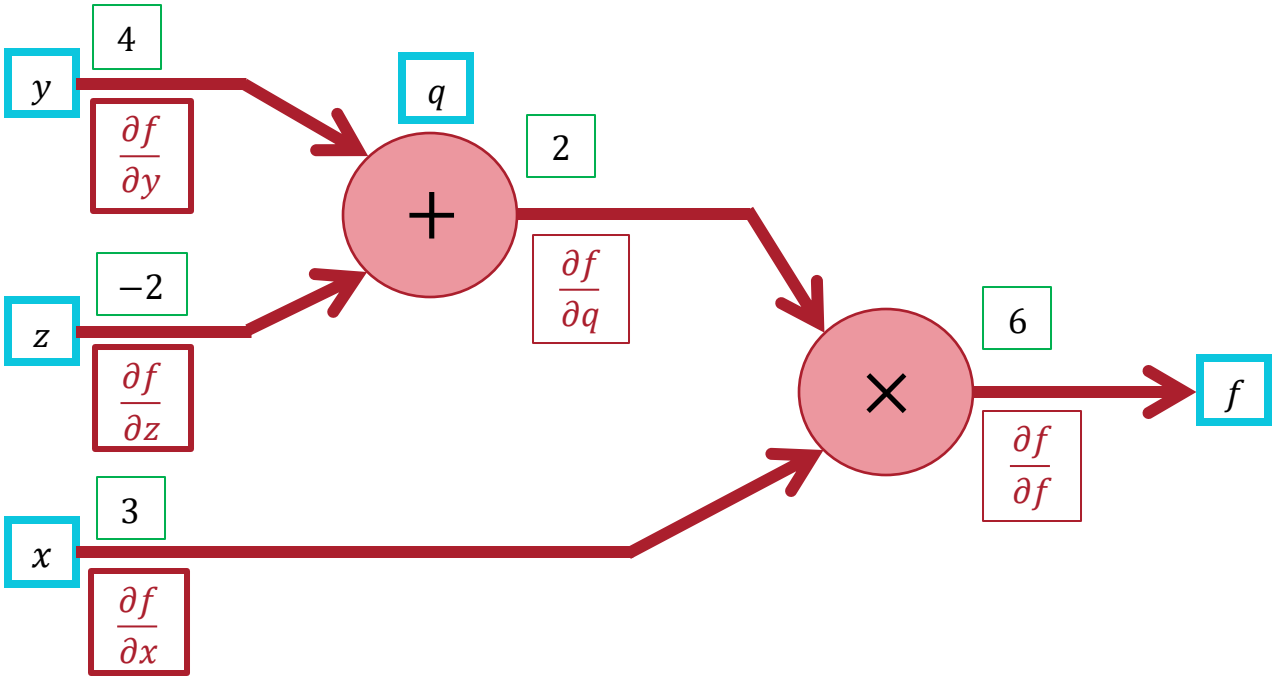


Check out the Computational Graph Worksheet if you'd like some extra practice!

Ask questions: menti.com
Code: 7201 0048

Backpropagation

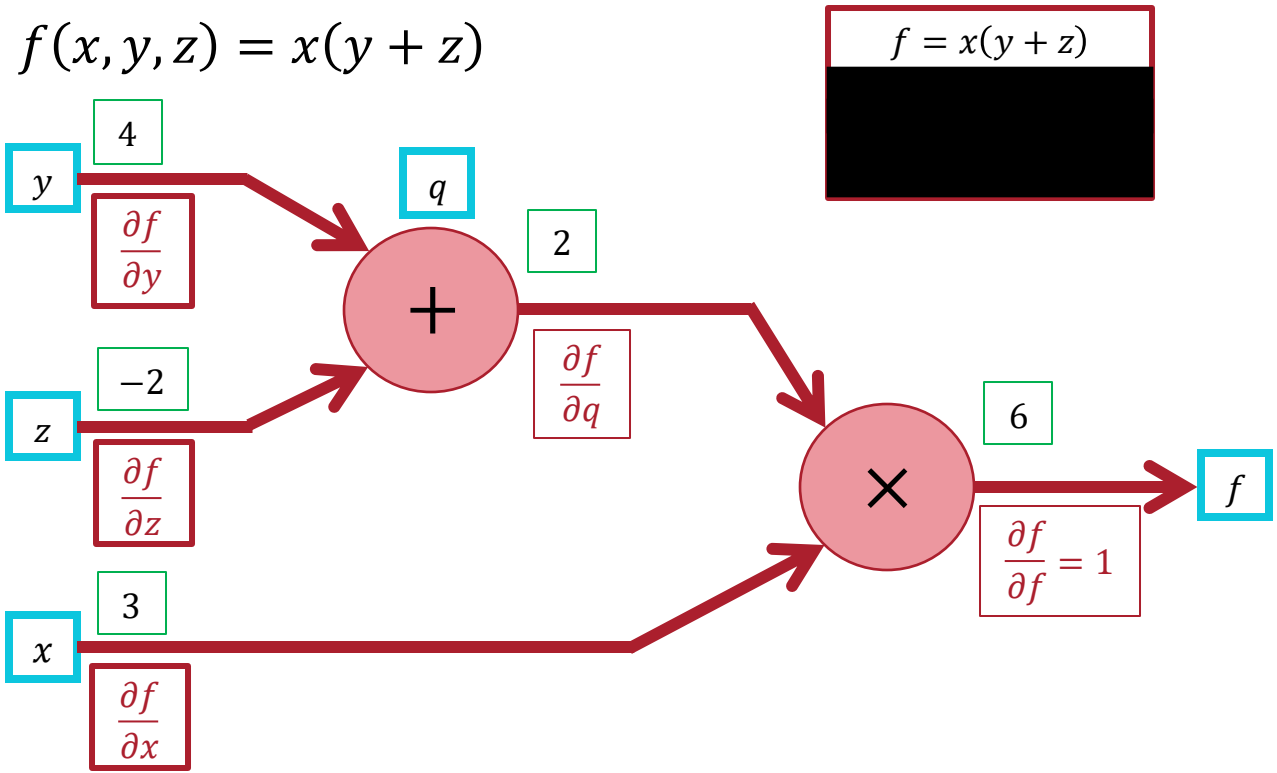
$$f(x, y, z) = x(y + z)$$



Ask questions: menti.com
Code: 7201 0048

Backpropagation

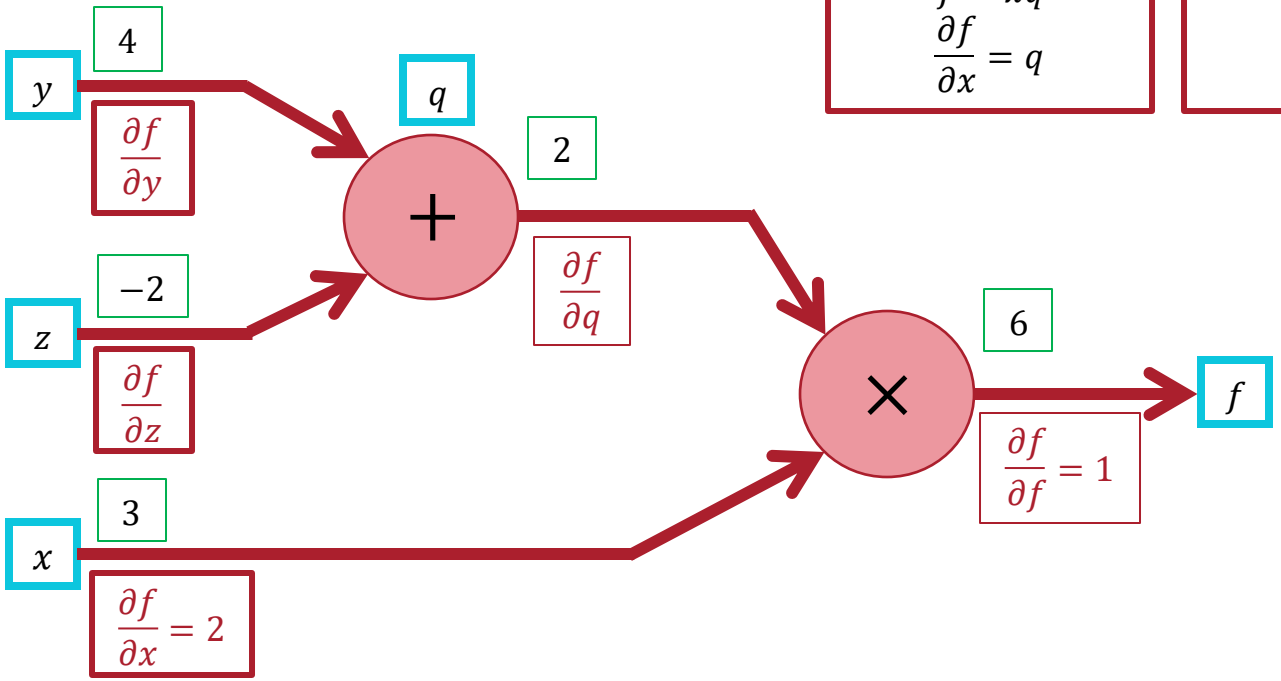
$f(x, y, z) = x(y + z)$



Ask questions: menti.com
Code: 7201 0048

Backpropagation

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



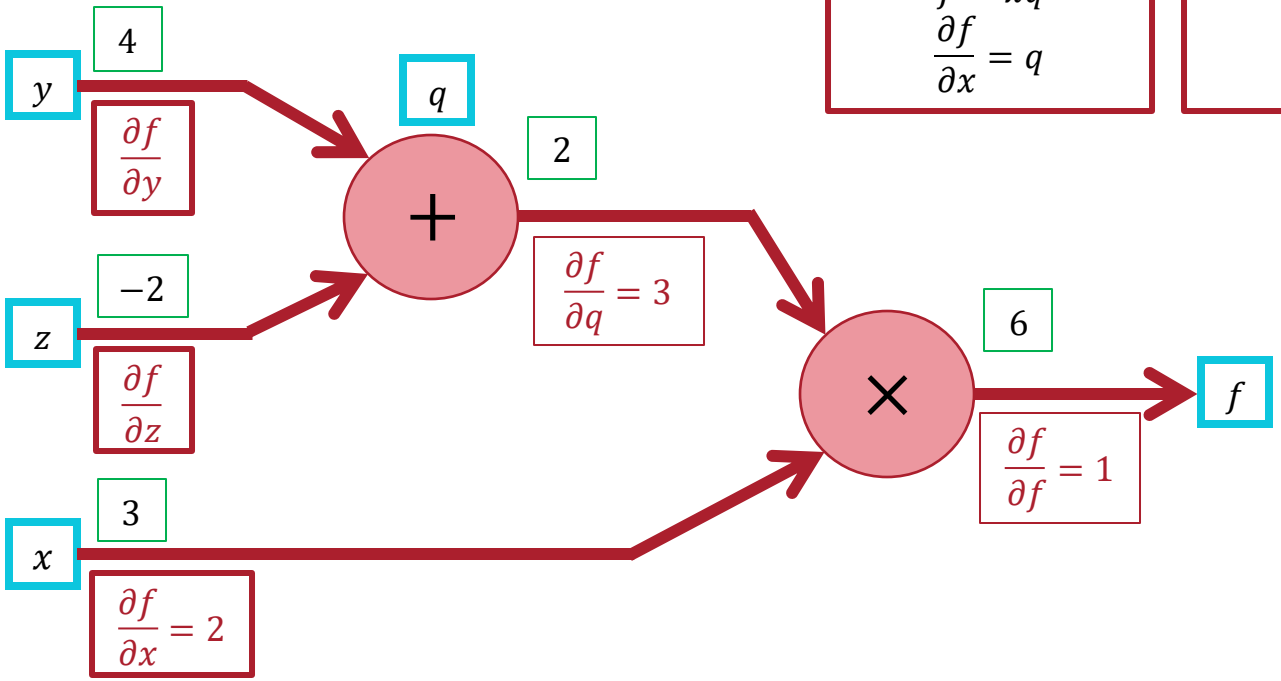
$f = x(y + z)$
$f = xq$
$\frac{\partial f}{\partial x} = q$

$f = x(y + z)$
$f = xq$
$\frac{\partial f}{\partial q} = x$

Ask questions: menti.com
Code: 7201 0048

Backpropagation

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



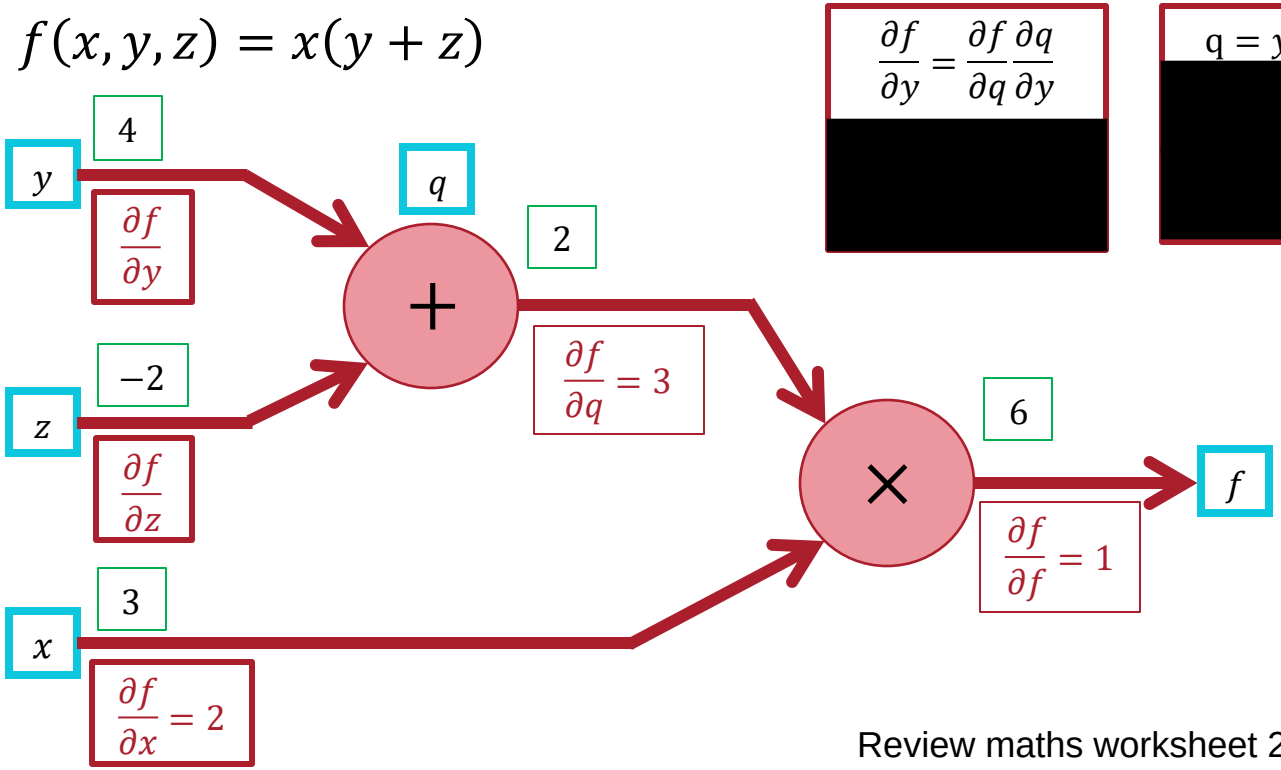
$f = x(y + z)$
$f = xq$
$\frac{\partial f}{\partial x} = q$

$f = x(y + z)$
$f = xq$
$\frac{\partial f}{\partial q} = x$

Ask questions: menti.com
Code: 7201 0048

Backpropagation

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

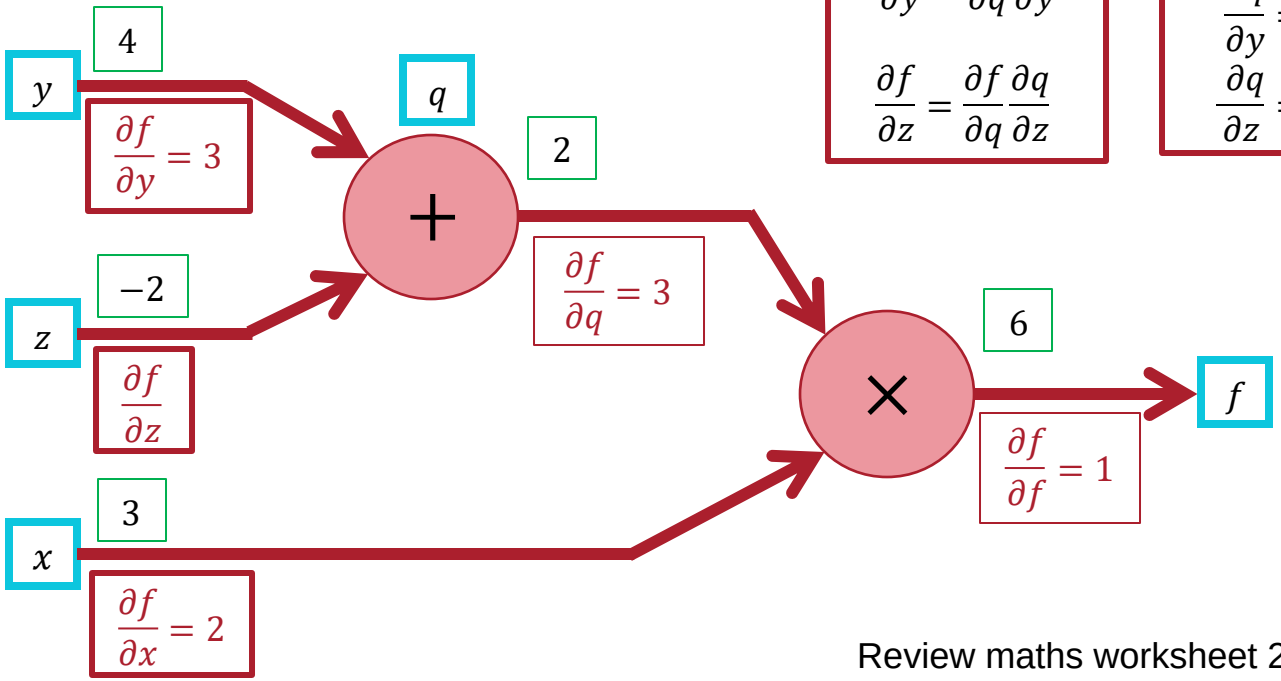


Review maths worksheet 2 if you are feeling rusty on the chain rule!

Ask questions: menti.com
Code: 7201 0048

Backpropagation

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



$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial y}$$

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial z}$$

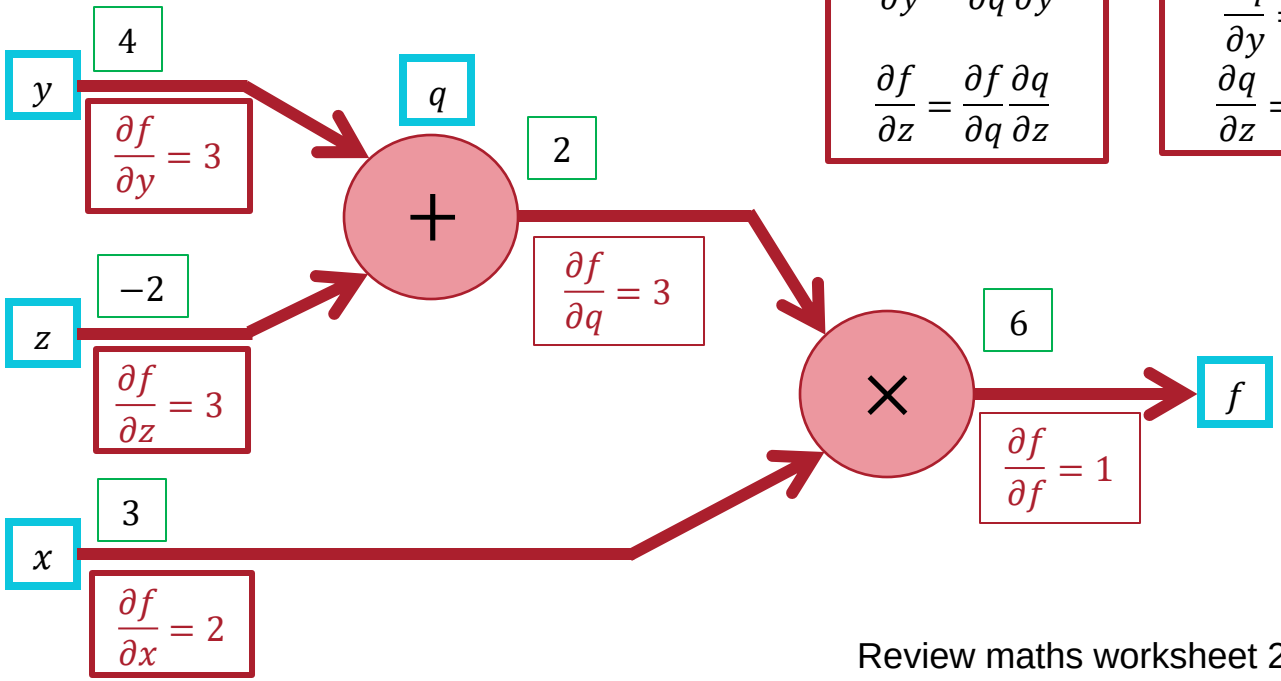
$$q = y + z$$
$$\frac{\partial q}{\partial y} = 1$$
$$\frac{\partial q}{\partial z} = 1$$

Review maths worksheet 2 if you are feeling rusty on the chain rule!

Ask questions: menti.com
Code: 7201 0048

Backpropagation

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



$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial y}$$

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial z}$$

$$q = y + z$$

$$\frac{\partial q}{\partial y} = 1$$

$$\frac{\partial q}{\partial z} = 1$$

Review maths worksheet 2 if you are feeling rusty on the chain rule!

Ask questions: menti.com
Code: 7201 0048

Your turn!

- Present your problem and answer to the class.

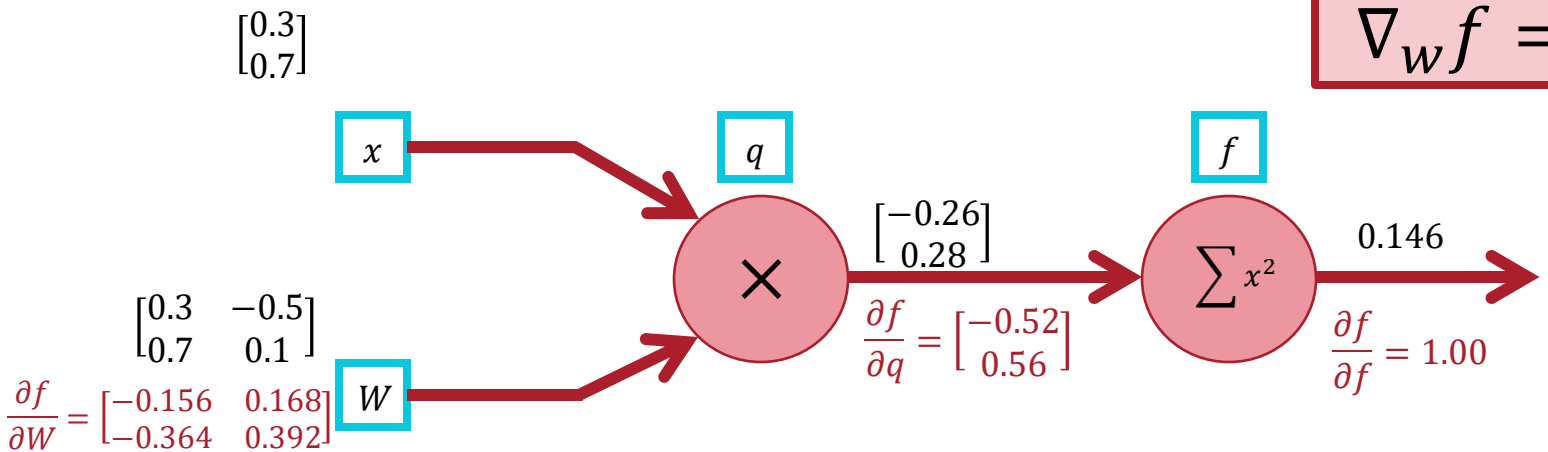
Problem Number	Groups
2	1, 4, 7, 10, 13
3	2, 5, 8, 11, 14
4	3, 6, 9, 12, 15

Ask questions: menti.com
Code: 7201 0048

Vectorised...

$$f(x, W) = \sum_{i=1}^{n} (W \cdot x)_i^2$$

$$\nabla_w f = 2q \cdot x^T$$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix} \longrightarrow \frac{\partial q_k}{\partial W_{i,j}} = x_j$$

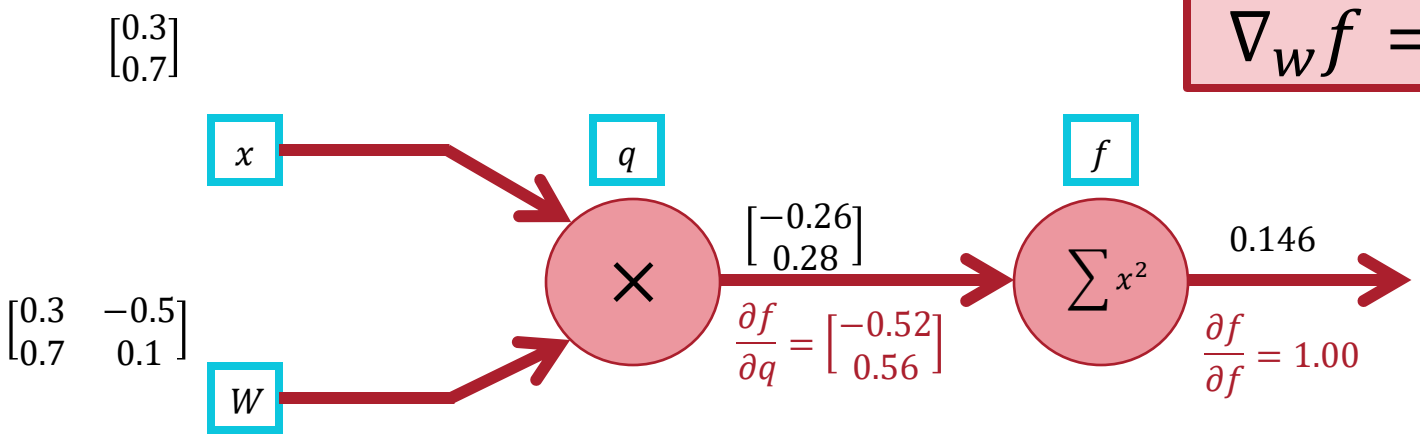
$$f(q) = \sum_{i=1}^{n} (q)_i^2 = q_1^2 + \cdots + q_n^2 \longrightarrow \frac{\partial f}{\partial q_i} = 2q_i \qquad \frac{\partial f}{dW_{i,j}} = \sum_k \frac{\partial f}{\partial q_k}\frac{\partial q_k}{\partial W_{i,j}} = \sum_k (2q_k)x_j = 2q_i x_j$$

Ask questions: menti.com
Code: 7201 0048

Vectorised...

$$f(x, W) = \sum_{i=1}^{n} (W \cdot x)_i^2$$

$$\nabla_w f = 2q \cdot x^T$$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix} \longrightarrow \frac{\partial q_k}{\partial W_{i,j}} = x_j$$

$$f(q) = \sum_{i=1}^{n} (q)_i^2 = q_1^2 + \cdots + q_n^2 \longrightarrow \frac{\partial f}{\partial q_i} = 2q_i$$

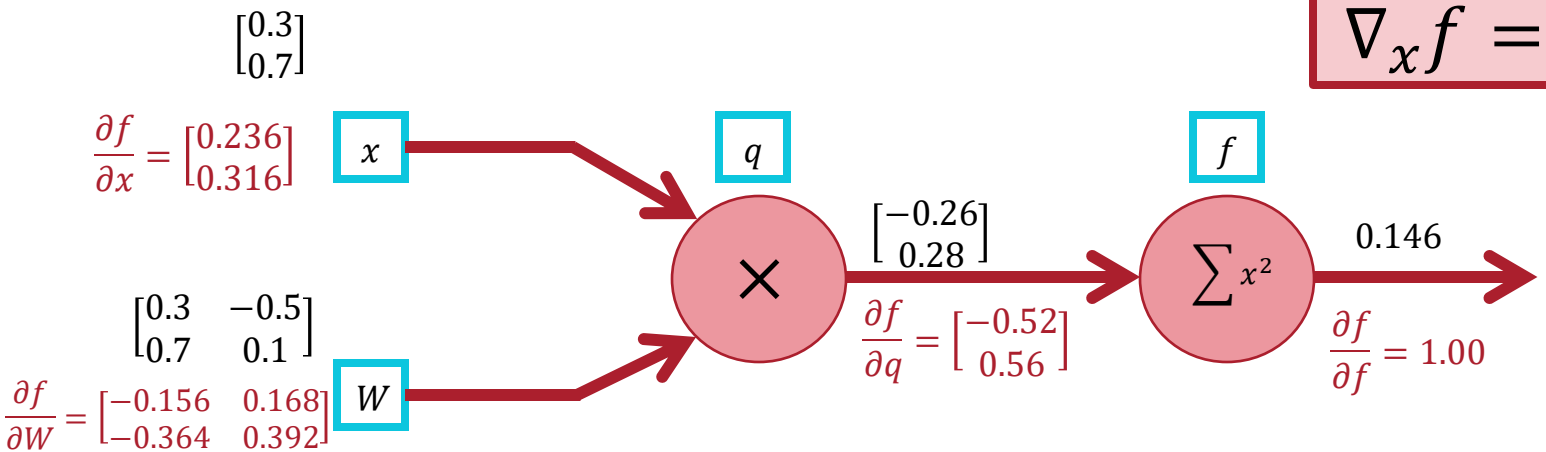
$$\frac{\partial f}{dW_{i,j}} = \sum_k \frac{\partial f}{\partial q_k}\frac{\partial q_k}{\partial W_{i,j}} = \sum_k (2q_k)x_j = 2q_i x_j$$

Ask questions: menti.com
Code: 7201 0048

Vectorised...

$$f(x, W) = \sum_{i=1}^{n} (W \cdot x)_i^2$$

$$\nabla_x f = W^T \cdot 2q$$



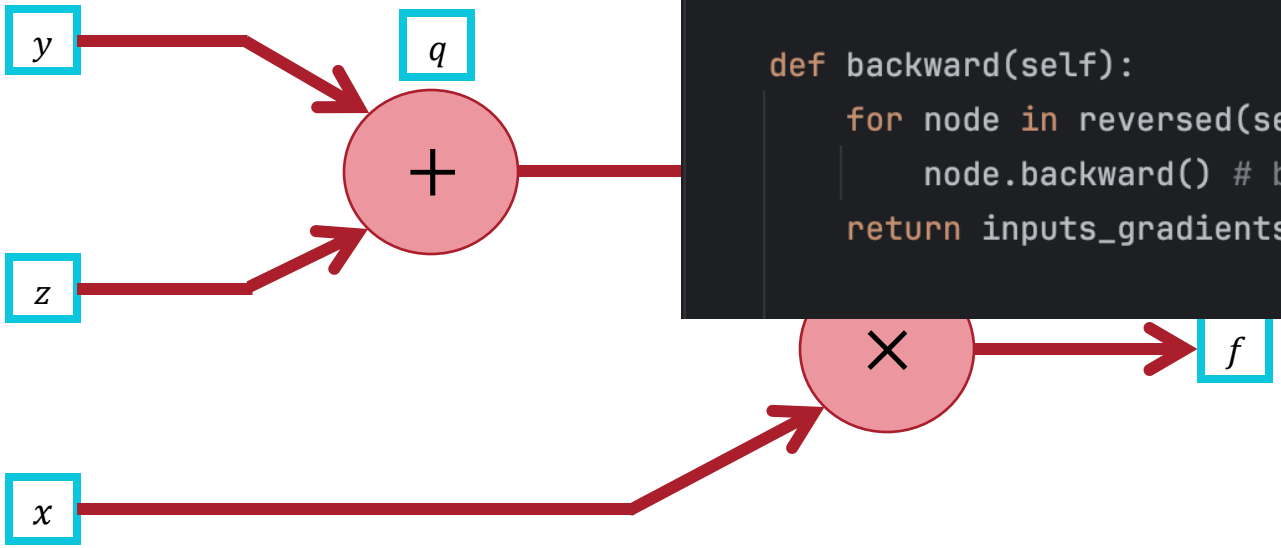
$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix} \longrightarrow \frac{\partial q_k}{\partial x_i} = W_{k,i}$$

$$f(q) = \sum_{i=1}^{n} (q)_i^2 = q_1^2 + \cdots + q_n^2 \longrightarrow \frac{\partial f}{\partial q_i} = 2q_i$$

$$\frac{\partial f}{dx_i} = \sum_k \frac{\partial f}{\partial q_k}\frac{\partial q_k}{\partial x_i} = \sum_k (2q_k)W_{k,i}$$

Coding

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



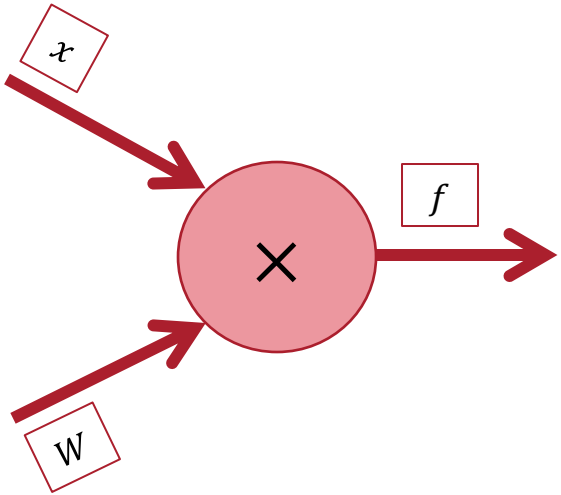

```python
class ComputationalGraph:

    def forward(self, inputs):
        for node in self.graph.nodes()
            node.forward()
        return loss

    def backward(self):
        for node in reversed(self.graph.nodes()):
            node.backward() # backpropagation
        return inputs_gradients
```

Coding

$$f(x, W) = Wx$$




```python
class MultiplyGate(object):
    def forward(self, x, W):
        self.x = x
        self.W = W
        output = self.x * self.W
        return output

    def backward(self, dz):
        # df/dx
        dx = self.W * dz # [dz/dx * df/dz]

        # df/dW
        dW = self.x * dx # [dx/dW * df/dz]

        return [dx, dy]
```

Application

- Optimisation

$$w_x^* = w_x - \alpha\left(\frac{\partial L}{\partial w_x}\right)$$

Loss Function, L ← Loss

Wx → $\begin{bmatrix} 145.8 \\ -259.6 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Backpropagation: where did it come from?

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533-536.

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

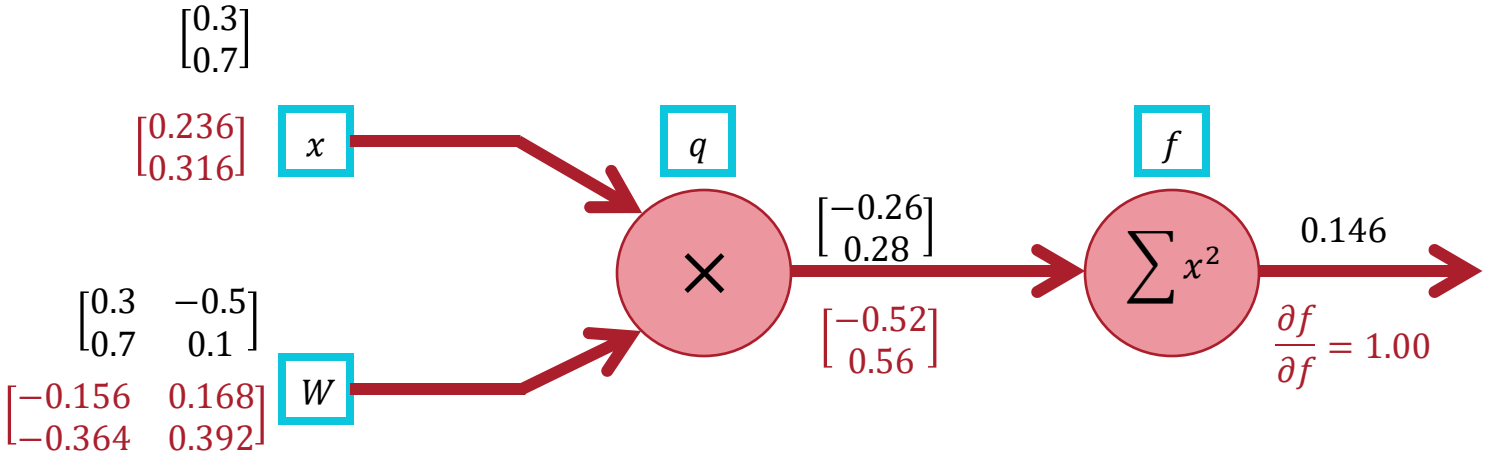
Ask questions: menti.com
Code: 7201 0048

Recommended Reading List

- Deep Learning, Section 6.5 Back-Propagation and Other Differentiation Algorithms (see Blackboard for pdf)
- Hinton, G. (2022). The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*.
- Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(153):1--43, 2018
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Submit answers to : menti.com
Code: 7201 0048

Drawbacks



Ask questions: menti.com
Code: 7201 0048

Drawbacks

Vanishing gradients: this becomes a real problem in deeper networks (i.e. those with more layers)

What if we can’t differentiate the function?

Not really biologically plausible

Hinton, G. (2022). The forward-forward algorithm: Some preliminary investigations. arXiv preprint arXiv:2212.13345.

Ask questions: menti.com
Code: 7201 0048

A Look Back At Our Objectives

Understand:

- What backpropagation *is*
  - *An algorithm that allows us (in combination with gradient descent) to efficiently train neural networks by minimising the error*
- How it *works*
  - *Compute loss, backpropagate layer by layer (then use gradient descent to update the weights)*
- Why it's *useful* to us
  - *Allows us to efficiently update the weights of the network*

University of BRISTOL

BEFORE YOU GO

Please could you indicate on the menti poll how well you feel you understand backpropagation on a scale of 1-5.

Scan the QR code to book a drop-in if you have struggled to understand this material.

bristol.ac.uk

University of BRISTOL

ANY QUESTIONS?

bristol.ac.uk