Backpropagation: way of computing gradients of expressions through recursive application of chain rule.

Simple expressions and interpretation of the gradient:

$f(x, y) = xy$

$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}] = [y, x]$

- The derivative on each variable tells you the sensitivity of the whole expression on its value.

Compound expressions with chain rule

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

Backpropagation: way of computing gradients of expressions through recursive application of chain rule.

Intuitive understanding of backpropagation: Backpropagation can thus be thought of as gates communicating to each other (through the gradient signal) whether they want their outputs to increase or decrease (and how strongly), so as to make the final output value higher.

Staged computation: break up your function into modules for which you can easily derive local gradients, and then chain them with chain rule. Crucially, you almost never want to write out these expressions on paper and differentiate them symbolically in full, because you never need an explicit mathematical equation for the gradient of the input variables. Hence, decompose your expressions into stages such that you can differentiate every stage independently (the stages will be matrix vector multiplies, or max operations, or sum operations, etc.) and then backprop through the variables one step at a time.