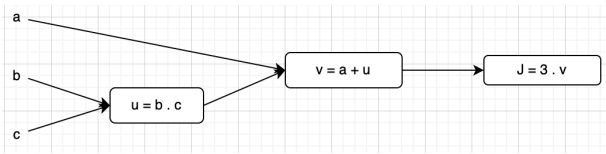
Computation graph

J (a, b, c) = 3(a + bc)

- u = b.c
- v = a + u
- J = 3.v

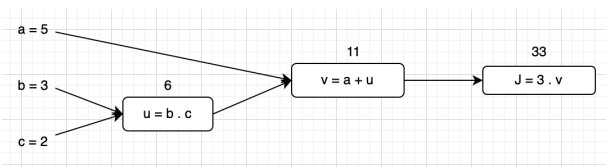


Computation graph comes in handy when there're some special output variable such as J above that you want to optimize.

In logistic regression, of course J is the cost function which we want to optimize.

Takeaway: through the left to right pass, we can compute the output value of J
through the right to left pass, we can compute the gradient or the derivative of J

Computing derivatives using computation graph



In the above diagram, what is the value of the derivate of J with respect to v (in other words: if we change the value of v a little bit, how would it affect the value of J?)

$$\frac{dJ}{dv} = ?$$

Apply the derivative rule, we have: $J = 3 \cdot v \Rightarrow \frac{dJ}{dv} = 3$

What is the value of the derivative of J with respect to a? (In other words words: If we change the value of a a little bit, how would it affect the value of J? Note that a is not directly affect the value of J but through v)

$$\frac{dJ}{da} = ?$$

First, by changing a, you ended up changing the value of v. How much does v change? It is change by an amount that's determined by dv/da. And then the change in v will cause the value of J to also increase.

-> In calculus, this is called the chain rule that if a affects v, affects j, then the amounts that J changes when you nudge a is the product of how much v changes when you nudge a times how much J changes when you nudge v

$$\frac{dJ}{da} = \frac{dJ}{dv} \cdot \frac{dv}{da}$$

When you're writing codes to implement back propagation, there will be some final output variables that you really care about. And in that case, the final output variable is J. It't the last node in our computation graph.

-> A lot of computations will be trying to compute the derivative of that final output variable with various intermediate variables, such as a, b, c, u or v.

Notations:

- In code, when you want to represent the value $\frac{d_{final\ output\ var}}{d_{var}}$ you should write code like "dvar" instead of writing the whole thing
- So dvar in a code you write will represent the derivative of the final output variable you care about such as J with respect to the various intermediate quantities.