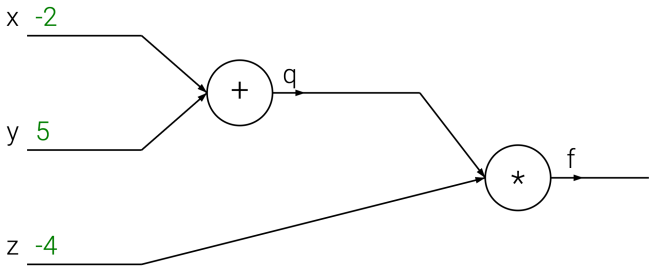
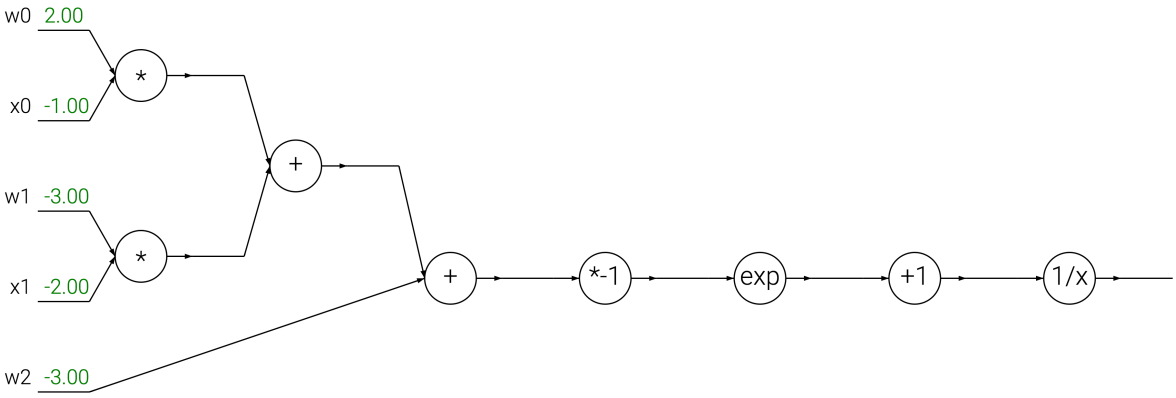
Backpropagation

(find and work with a partner)

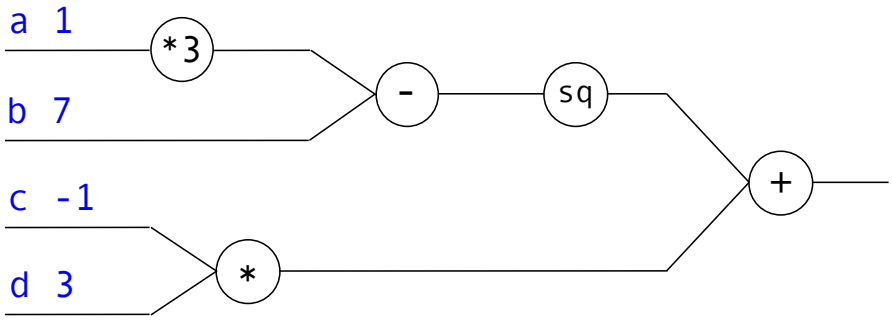
1. Let $f(x, y, z) = (x + y)z$, a function of three inputs. Let $q = x + y$, so we can rewrite this function as $f = qz$. To determine how f changes as each input changes, we will use backpropagation through this neural network. First run the "forward pass" to compute the output value of each node (write above the lines). Then use the idea of the chain rule to compute the derivative of each node with respect to the inputs (write below the lines).



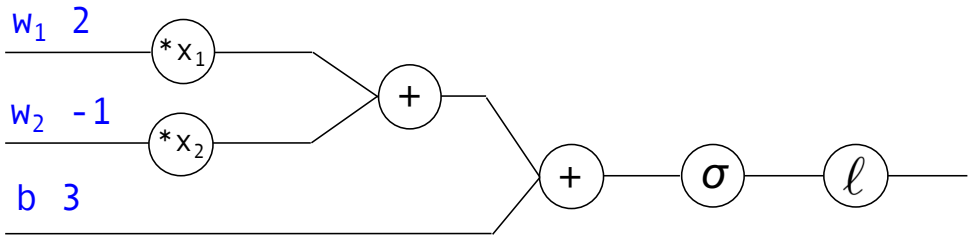
2. Let $f(w_0, w_1, w_2, x_0, x_1) = \frac{1}{1+e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$. Again compute the forward pass (can use a calculator) to determine the output value of each node, then use backpropagation to determine the gradients.



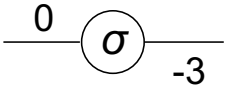
3. Let $f(a, b, c, d) = (3a - b)^2 + cd$, a function of four inputs. We could decompose the function into a neural network diagram in many ways, but one way is shown below. Using the given values, first run the "forward pass" to compute the output value of each node (write above the lines). Then use backpropagation to compute the derivatives (write below the lines). If we wanted to *minimize* f with respect to these inputs and we have a learning rate of $\alpha = 0.1$, what values of a, b, c, d would we choose for the next step?



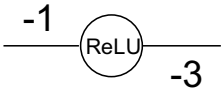
4. Let $\sigma(z) = \frac{1}{1+e^{-z}}$ be the sigmoid function, and let $h(\vec{x}) = \sigma(w_1x_1 + w_2x_2 + b)$ be our hypothesis about the label (0 or 1) of a given input vector $\vec{x} = (x_1, x_2)$ with two features. Let our loss function be the cross entropy of our prediction relative to the truth, i.e. $\ell_y(h) = -y \log h - (1-y)\log(1-h)$. If we are given fixed values $\vec{x} = (1, 3)$ and label $y = 0$, what is the value of the loss function (given the starting weights below)? Again use backpropagation to compute the gradients. Use the fact that $\sigma'(z) = \sigma(z)(1 - \sigma(z))$. Assuming we want to minimize the loss and $\alpha = 0.1$, what are the new values of w_1, w_2, b?



5. Say that we are using sigmoid as our non-linearity, and the input coming into the activation function is 0. If the gradient flowing back into sigmoid is -3, what gradient value will sigmoid pass along to the previous unit? In other words, fill in the missing entries in this backpropagation diagram:



6. Say we are using ReLU as our non-linearity, and the input coming into the activation function is -1. If the gradient flowing back into ReLU is -3, what gradient value will ReLU pass along to the previous unit?



7. Demonstrate that tanh is a rescaling of the sigmoid function, specifically:

$$\tanh(x) = 2\sigma(2x) - 1$$

Acknowledgements: first examples from Stanford course CS231n: http://cs231n.github.io/optimization-2/