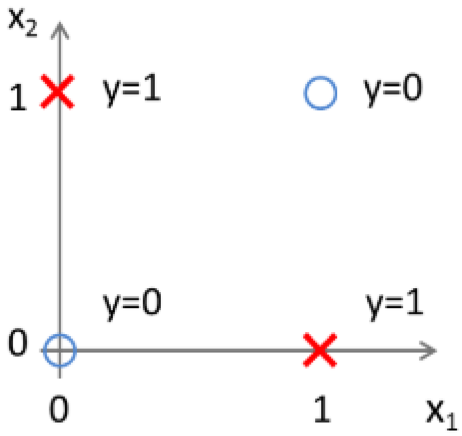
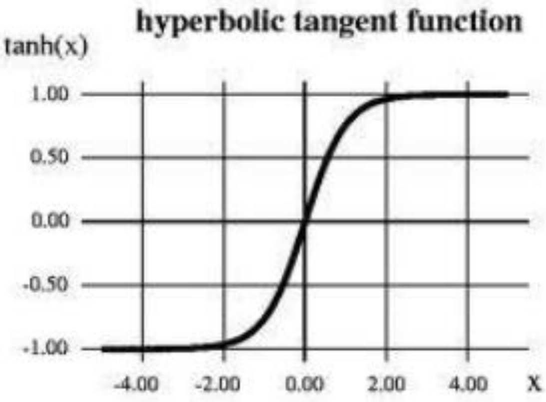
# Lab 5 - Neural Networks

cs542- Spring 2020

# 1.Neural Network for XOR

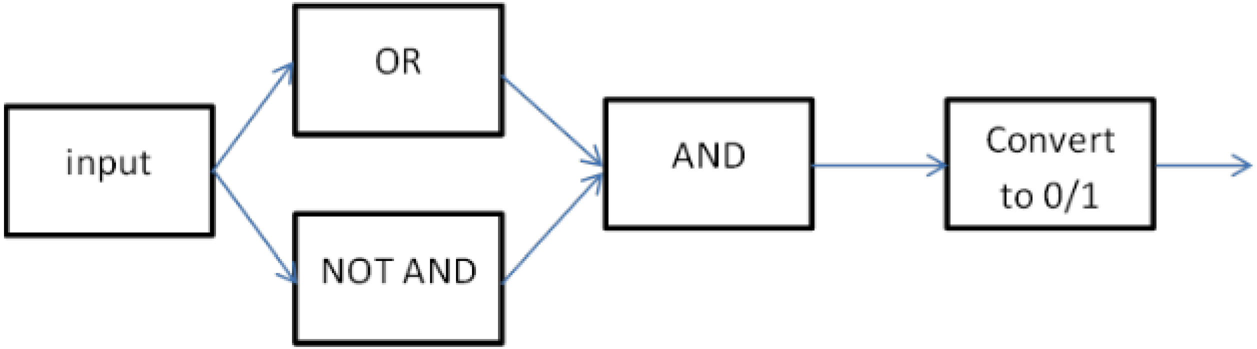
Design a neural network to solve the XOR problem, i.e. the network should output 1 if only one of the two binary input variables is 1, and 0 otherwise (see left figure). Use the hyperbolic tangent, or *tanh,* activation function in all nodes (right figure), which ranges in [-1, +1].



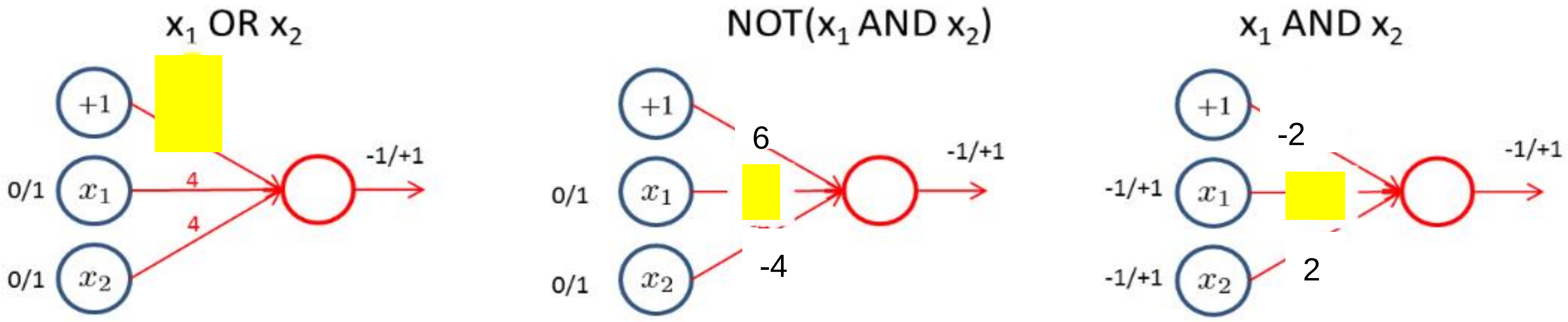



# Answer

Note that (A XOR B) can be expressed as (A OR B) AND NOT(A AND B)), as illustrated below:

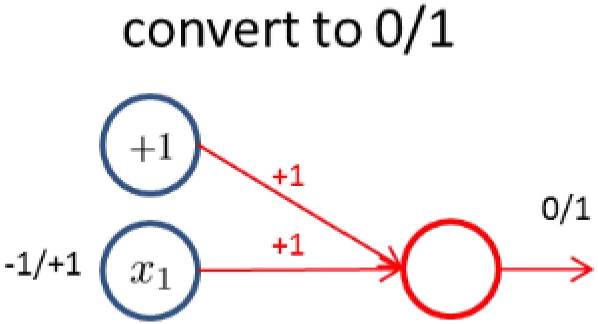


In the diagrams below, we filled in most of the tanh units' parameters. Fill in the remaining parameters, keeping in mind that tanh outputs +1/-1, not 0/1. Note that we need to appropriately change the second layer (the AND node) to take +1/-1 as inputs. Also, we must add an extra last layer to convert the final output from +1/-1 to 0/1. *Hint: assume tanh outputs* $-1$ *for any input* $x \leq -2$, $+1$ *for any input* $x \geq +2$, $0$ *for* $x = 0$.

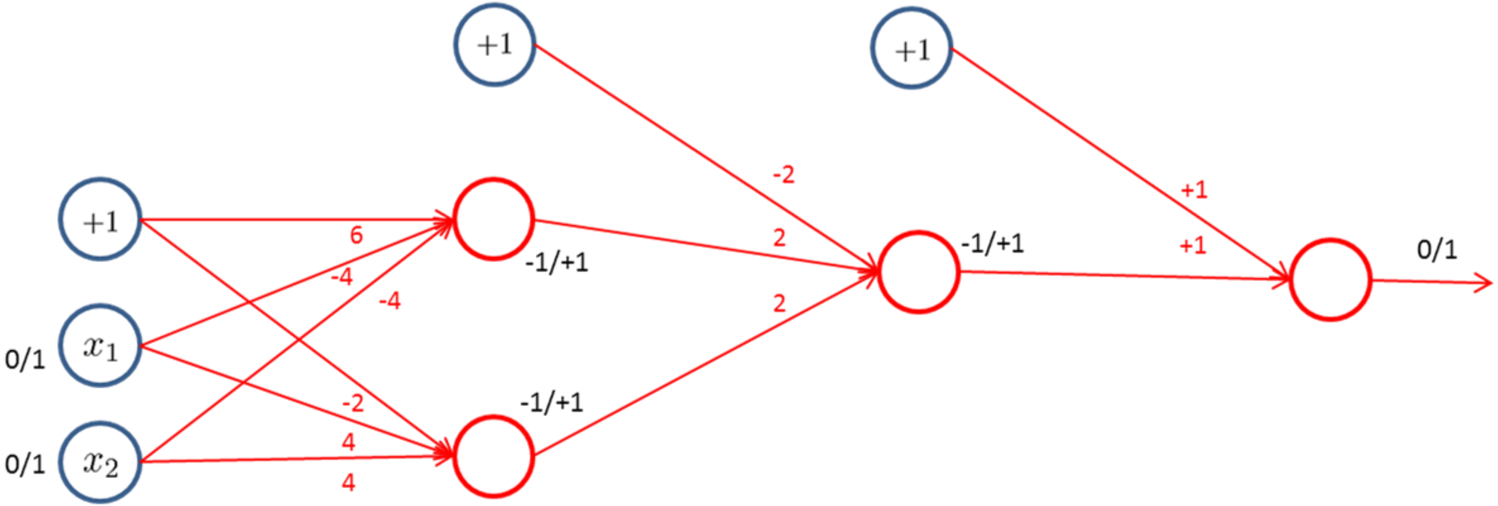



Answer: -2, -4, +2

# Convert input x1ϵ{-1,1} to 0/1



# Final XOR network



# 2. Computation Graph and Backpropagation

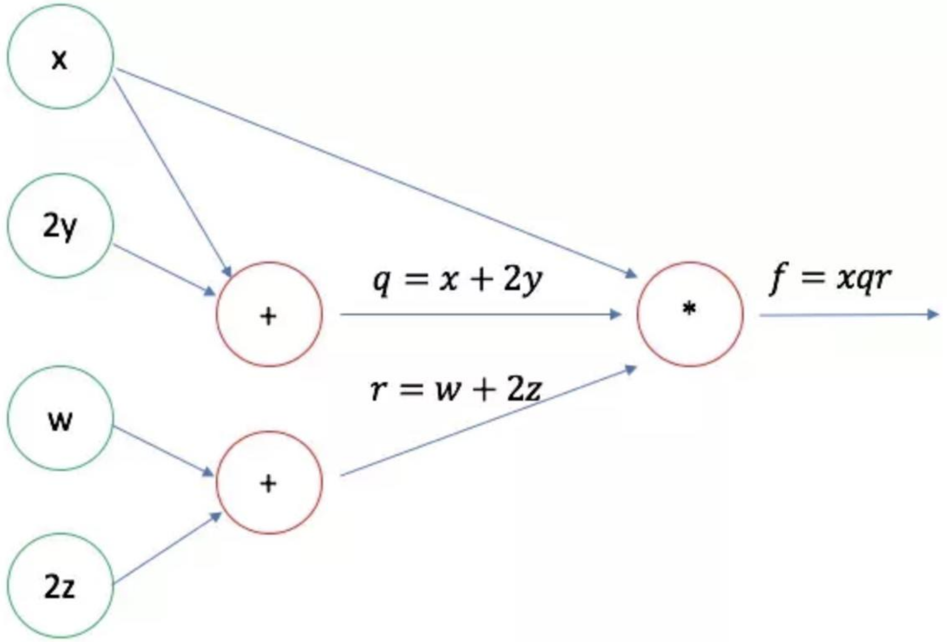
In class, we learned how to take a complex function that consists of multiple nested functions and represent it with a computation graph, which allows us to write down the forward and backward pass used to compute the function gradient.

Practice converting functions to their computation graphs

$$f(w, x, y, z) = x(x + 2y)(w + 2z).$$

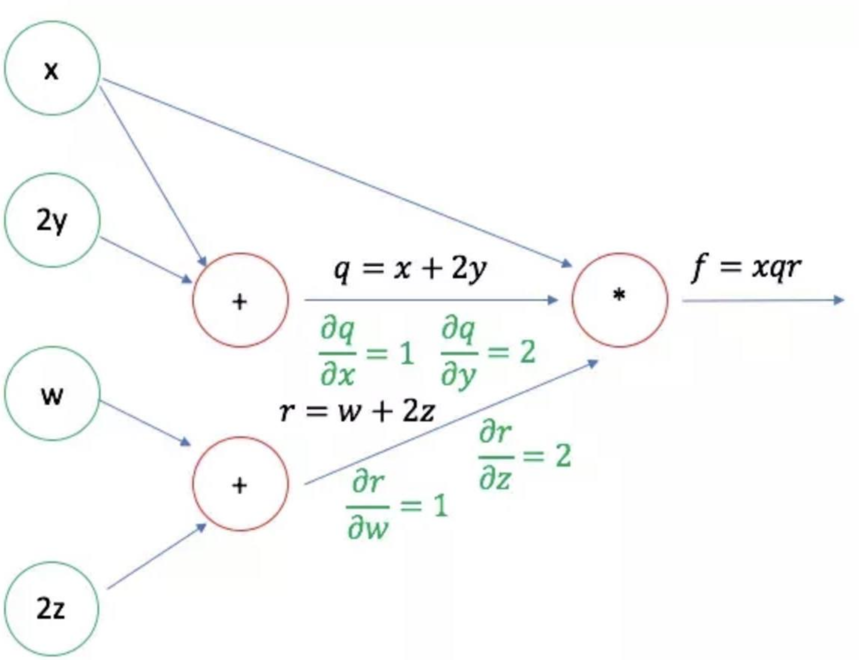
$$f(w, x, y, z) = x(x + 2y)(w + 2z).$$

**Computational Graph:**

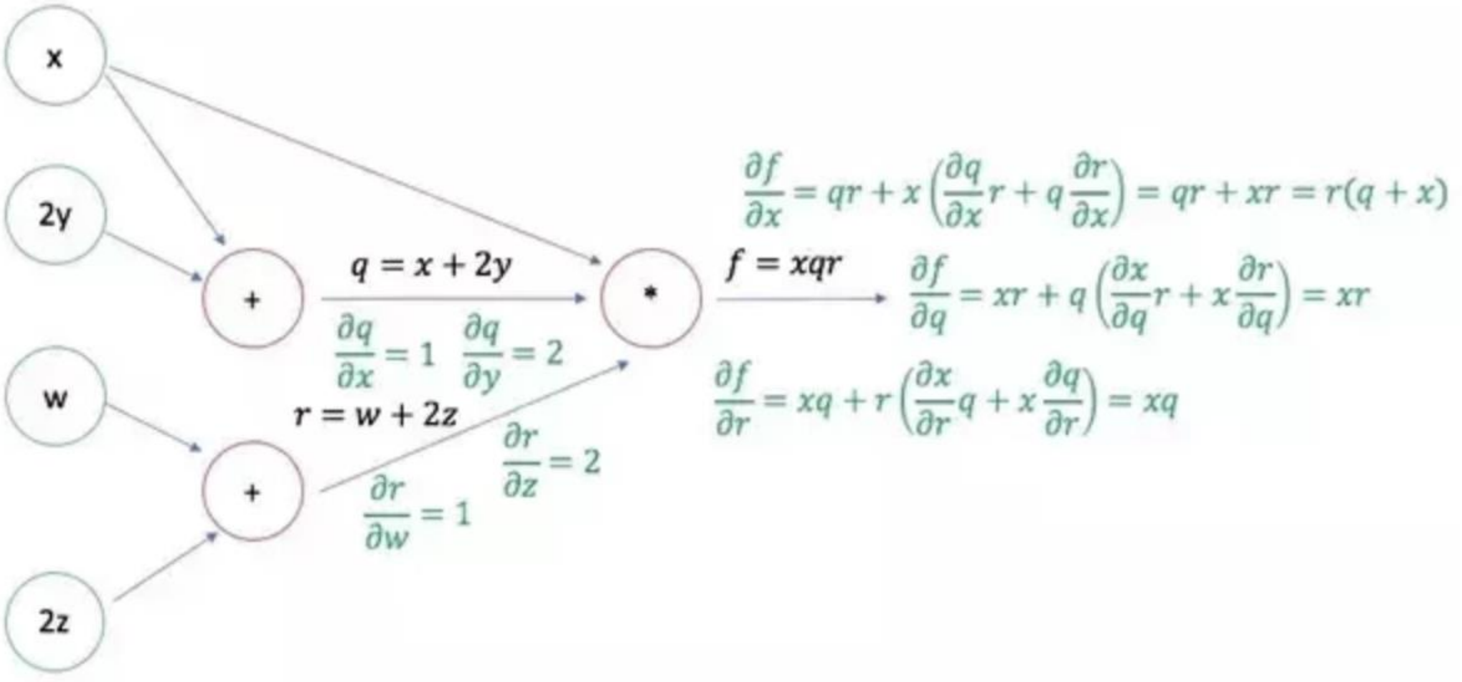


# For the computational graph obtained, write down the forward pass and the backward pass equations

(i)

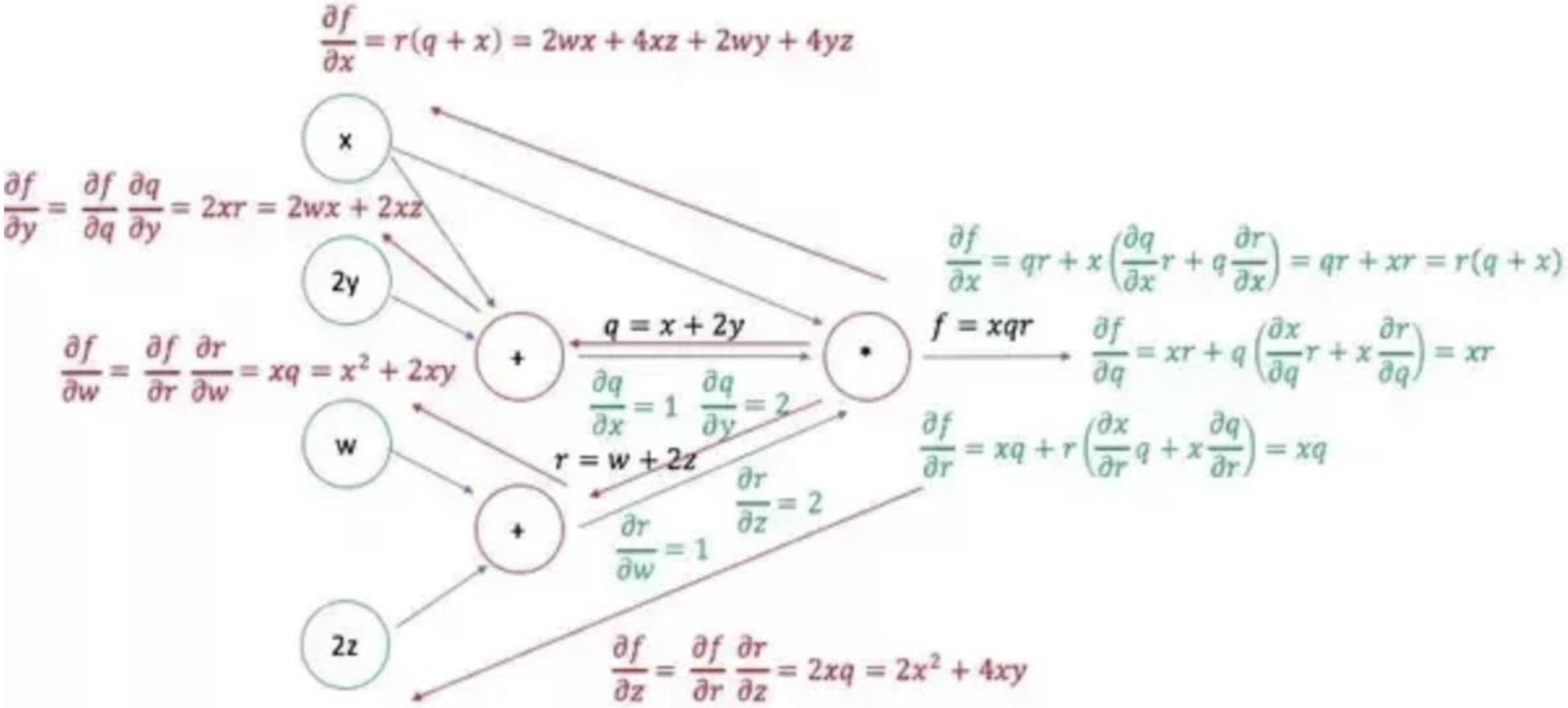


(ii)



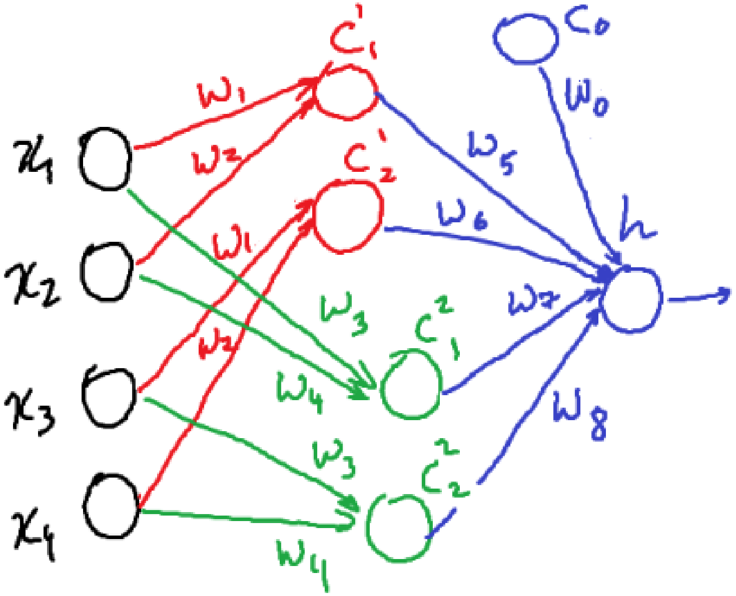
x
2y
w
2z
+
+
*
$q = x + 2y$
$\frac{\partial q}{\partial x} = 1$ $\frac{\partial q}{\partial y} = 2$
$r = w + 2z$
$\frac{\partial r}{\partial z} = 2$
$\frac{\partial r}{\partial w} = 1$
$f = xqr$
$\frac{\partial f}{\partial x} = qr + x\left(\frac{\partial q}{\partial x}r + q\frac{\partial r}{\partial x}\right) = qr + xr = r(q + x)$
$\frac{\partial f}{\partial q} = xr + q\left(\frac{\partial x}{\partial q}r + x\frac{\partial r}{\partial q}\right) = xr$
$\frac{\partial f}{\partial r} = xq + r\left(\frac{\partial x}{\partial r}q + x\frac{\partial q}{\partial r}\right) = xq$

(iii)


$\frac{\partial f}{\partial x} = r(q+x) = 2wx + 4xz + 2wy + 4yz$
x
2y
w
2z
+
+
•
$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial y} = 2xr = 2wx + 2xz$
$\frac{\partial f}{\partial w} = \frac{\partial f}{\partial r}\frac{\partial r}{\partial w} = xq = x^2 + 2xy$
$q = x + 2y$
$\frac{\partial q}{\partial x} = 1 \quad \frac{\partial q}{\partial y} = 2$
$r = w + 2z$
$\frac{\partial r}{\partial z} = 2$
$\frac{\partial r}{\partial w} = 1$
$f = xqr$
$\frac{\partial f}{\partial x} = qr + x\left(\frac{\partial q}{\partial x}r + q\frac{\partial r}{\partial x}\right) = qr + xr = r(q+x)$
$\frac{\partial f}{\partial q} = xr + q\left(\frac{\partial x}{\partial q}r + x\frac{\partial r}{\partial q}\right) = xr$
$\frac{\partial f}{\partial r} = xq + r\left(\frac{\partial x}{\partial r}q + x\frac{\partial q}{\partial r}\right) = xq$
$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial r}\frac{\partial r}{\partial z} = 2xq = 2x^2 + 4xy$

# 3. Neural Network Architectures

a) Draw a convolutional network with input $x \in R^4$, one hidden layer with 2x1 filters and 2 channels with stride 2, and a fully-connected output layer with one neuron.




Answer: 9 parameters, see w's in the plot. The c's are the hidden convolutional units, with each channel sharing parameters. Note that the last layer should also have a "dummy" unit set to 1 and thus an extra parameter.

b) What algorithm is used for learning the parameters of a recurrent network? Name the algorithm and sketch out its main steps.

Answer: Backpropagation in time. The steps for computing the gradient for 1 training sequence are as follows. First, we unroll the network by replicating it once for each time step in our training sequence. Then we use regular backprop on the unrolled network to find the gradient.