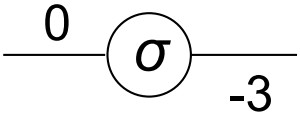
Neural Networks, short questions *(find and work with a partner)*

1. *Activation Functions.* So far we have seen three different activation functions:

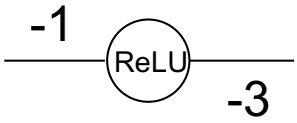
   - Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$
   - Hyperbolic Tangent: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
   - ReLU: $f(x) = \max(0, x)$

   (a) If the input has value exactly 0, what is the activation of each function above?

   (b) 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:



   (c) 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?



   (d) (for after class) Demonstrate that tanh is a rescaling of the sigmoid function, specifically:

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

2. *Cross-Entropy Loss.* Last time we defined cross-entropy between two discrete probability distributions p and q (with the same set of inputs $\mathcal{X}$) as

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log_2 q(x).$$

When using cross-entropy as a classification loss function between the true one-hot class label vector y and the output probabilities for each class $\hat{y}$, we have

$$H(y, \hat{y}) = -\sum_{k=1}^{C} y_k \log_2 \hat{y}_k,$$

where C is the number of classes. Say we have $C = 3$. For a given data point $\vec{x}$, say the true label is class 2, and our neural network produced the probabilities $\hat{y} = [\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]$. What is the cross-entropy loss for this data point?

If instead our method produced probabilities $\hat{y} = [\frac{3}{8}, \frac{1}{8}, \frac{1}{2}]$, what is the loss for this data point?

Note: for multiple examples, we usually use the loss function:

$$J(\text{all weights}) = \frac{1}{n} \sum_{i=1}^{n} H(y_i, \hat{y}_i)$$

3. *Neural Network Architectures.* Say we have a fully connected neural network with 3 ~~hidden~~ layers. The input data has shape $(n, p) = (100, 3)$, the first and second hidden layers each have 4 units (i.e. $p_1 = p_2 = 4$), and we have one output. If we use biases for all 3 layers, how many total parameters do we need to optimize?