1. Recall that the softmax function takes in a vector $(z_1, \cdots, z_D)$ and returns a vector $(y_1, \cdots, y_D)$. We can express it in the following form:

$$r = \sum_j e^{z_j} \qquad y_i = \frac{e^{z_i}}{r}.$$

   (a) Consider D = 2, i.e. just two inputs and outputs to the softmax. Draw the computation graph relating z_1, z_2, r, y_1, and y_2.

   (b) Determine the backprop updates for computing the error signal $\bar{z}_j$ when given the error signal $\bar{y}_i$. You do not need to justify your answer.

   (c) Write a function to implement the vector-Jacobian product (VJP) for the softmax function based on your answer from part (b). For efficiency, it should operate on a mini-batch. The inputs are:

      - a matrix Z of size $N \times D$ giving a batch of input vectors. N is the batch size and D is the number of dimensions. Each row gives one input vector z = $(z_1, \cdots, z_D)$.
      - A matrix $\bar{Y}$ giving the output error signals. It is also $N \times D$.

   The output should be the error signal Z_{bar}. Do not use a for loop.

2. Recall the logistic activation function σ and the tanh activation function:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \qquad tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

   Both activation functions have a sigmoidal shape.

   (a) Give the Jacobian matrix $\partial \boldsymbol{y} / \partial \boldsymbol{z}$ of the tanh activation function, applied element-wise to all of the units in a layer. You may give your answer in terms of tanh'(z), the univariate derivative of the tanh function.

   (b) One of the difficulties with the logistic activation function is that of saturated units. Briefly explain the problem, and whether switching to tanh fixes the problem. (You may refer to your answer from part (a) or sketch the activation functions.)

   (c) Briefly explain one way in which using tanh instead of logistic activations makes optimization easier.

3. Consider a layer of a multilayer perceptron which has ReLU activations:

$$\begin{aligned} z_i &= \sum_j X_j w_{ij} x_j + b_i && (1) \\ h_i &= ReLU(z_i) && (2) \end{aligned}$$

(a) Give the backprop rules for computing the error signals z_i, x_j and w_{ij} in terms of the error signals h_i.

(b) Consider a pair of units (x_j, h_i). Based on your answer to part (a), for what values of x_j and z_i are we guaranteed that $w_{ij} = 0$?