1. Name three popular activation functions. Can you draw them?
   Popular activation functions include the step function, the sigmoid function, the hyperbolic tangent (tanh) function, and the Rectified Linear Unit (ReLU) function
2. Suppose you have an MLP composed of one input layer with 10 passthrough neurons, followed by one hidden layer with 50 artificial neurons, and finally one output layer with 3 artificial neurons. All artificial neurons use the ReLU activation function

    1. What is the shape of the input matrix **X**?
    2. What are the shapes of the hidden layer's weight matrix **W*h*** and bias vector **b*h***?
    3. What are the shapes of the output layer's weight matrix **W*o*** and bias vector **b*o***?
    4. What is the shape of the network's output matrix **Y**?
    5. Write the equation that computes the network's output matrix **Y** as a function of **X**, **W*h***, **b*h***, **W*o***, and **b*o***.

    1. The shape of the input matrix **X** is $m \times 10$, where m represents the training batch size.
    2. The shape of the hidden layer's weight matrix $\mathbf{W}_h$ is 10×50, and the length of its bias vector $\mathbf{b}_h$ is 50.
    3. The shape of the output layer's weight matrix $\mathbf{W}_o$ is 50×3, and the length of its bias vector $\mathbf{b}_o$ is 3.
    4. The shape of the network's output matrix **Y** is $m \times 3$.
    5. $\mathbf{Y} = \text{ReLU}(\text{ReLU}(\mathbf{X}\ \mathbf{W}_h + \mathbf{b}_h)\ \mathbf{W}_o + \mathbf{b}_o)$. Recall that the ReLU function just sets every negative number in the matrix to zero. Also note that when you are adding a bias vector to a matrix, it is added to every single row in the matrix, which is called *broadcasting*.

3. How many neurons do you need in the output layer if you want to classify email into spam or ham? What activation function should you use in the output layer? If instead you want to tackle MNIST, how many neurons do you need in the output layer, and which activation function should you use? What about for getting your network to predict housing prices.

To classify email into spam or ham, you just need one neuron in the output layer of a neural network—for example, indicating the probability that the email is spam. You would typically use the sigmoid activation function in the output layer when estimating a probability. If instead you want to tackle MNIST, you need 10 neurons in the output layer, and you must replace the sigmoid function with the softmax activation function, which can handle multiple classes, outputting one probability per class. If you want your neural network to predict housing prices, then you need one output neuron, using no activation function at all in the output layer. Note: when the values to predict can vary by many orders of magnitude, you may want to predict the logarithm of the target value rather than the target value directly. Simply computing the exponential of the neural network's output will give you the estimated value (since $\exp(\log v) = v$).