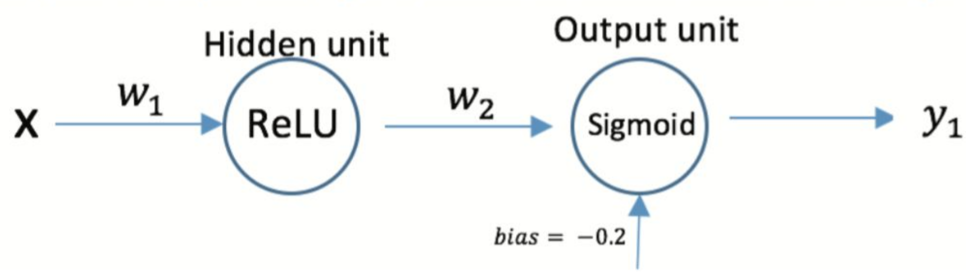
PROBLEM 25

1 ReLU Backpropagation**

1.1 Single output network

The rectified linear unit (ReLU) is a popular activation function for hidden layers. The activation function is a ramp function $f(z) = \max(0, z)$ where $z = wx$. This has the effect of simply thresholding its input at zero. Unlike the sigmoid, it does not saturate near 1 and is also simpler in gradient computations, resulting in faster convergence of SGD. Furthermore, ReLUs can allow networks to find sparse representations, due to their thresholding characteristic, whereas sigmoids will always generate non-zero values. However, ReLUs can have zero gradient when the activation is negative, blocking the backpropagation of gradients.

Here you use a very small neural network: it has one input unit, taking in a value x, one hidden unit (ReLU), and one output unit (sigmoid). We include a bias term of -0.2 on the sigmoid unit.



We use the following quantities in this problem:

$$z_1 = w_1 x$$

$$a_1 = \mathrm{ReLU}(z_1)$$

$$z_2 = w_2 a_1 - 0.2$$

$$y = \sigma(z_2)$$

The weights are initially $w_1 = \frac{1}{10}$ and $w_2 = -1$.

Let's consider one training example. For that training case, the input value is $x = 2$ (as shown in the diagram), and the target output value $t = 1$. We're using the following loss function:

$$E = \frac{1}{2}(y - t)^2$$

Please supply numeric answers; the numbers in this question have been constructed in such a way that you don't need a calculator. Show your work in case of mis-calculation in earlier steps.

(a) What is the output of the hidden unit for this input?

(b) What is the output of the output unit for this input?

(c) What is the loss, for this training example?

(d) Write out an abstract symbolic expression for derivative of the loss with respect to w_1 as repeated applications of the chain rule. For example, for the derivative of the loss with respect to w_2, we would write $\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial z_2}\frac{\partial z_2}{\partial w_2}$.

(e) Write the expression for each partial derivative in the chain rule expansion from the previous part. For example, $\frac{\partial y}{\partial z_2} = y(1-y)$.

(f) What is the derivative of the loss with respect to w_1, for this training example?

(g) What would the update rule for w_1 be?

(h) If η is large enough, w_1 will update from its current value of 0.1 to a negative value. Assume our new value is $w_1 = -0.1$. What will be the output of the output unit for an input of $x = 2$?

(i) What will happen when we try to update the weight, using this new example, for w_1 for any value of target? Why?

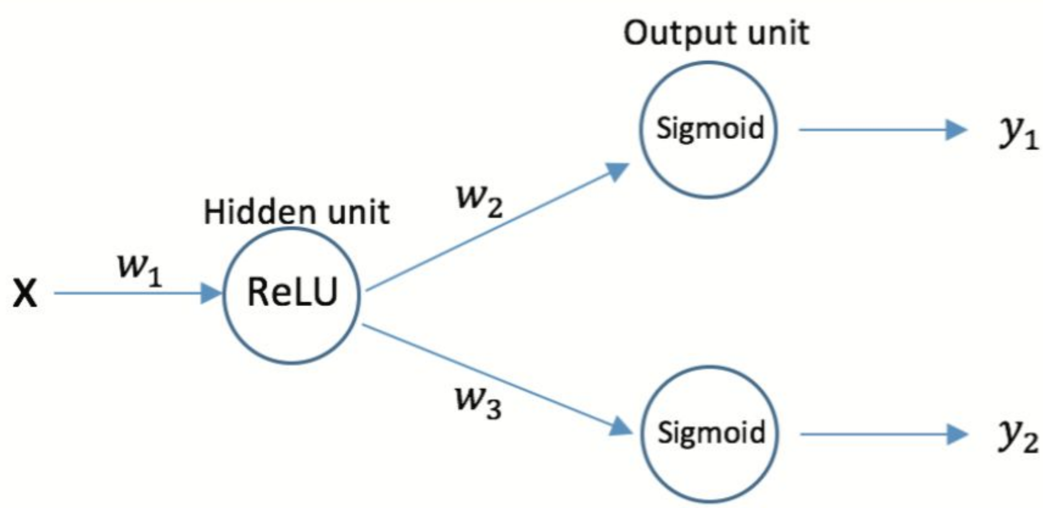
(j) Is it a bad idea to have a ReLU activation at the output layer?

(k) Consider the following activation function:

$$f(z) = \begin{cases} z & \text{if } z > 0 \\ \alpha z & \text{if otherwise.} \end{cases}$$

for some small alpha, e.g. $\alpha = 0.01$, and $z = wx$. Does this address the problem of dying ReLUs?

1.2 Multiple output network



(l)

$$a_1 = \mathrm{ReLU}(0, w_1 x)$$
$$y_1 = \sigma(w_2 a_1)$$

$$y_2 = \sigma(w_3 a_1)$$

Write out an abstract symbolic expression for the derivative of the loss with respect to w_1 for the network above with two output units, as repeated applications of the chain rule.

Multi-output (multi-class) networks are used in many settings such as object recognition, where we are trying to classify an image as being one of K objects. Each of the K possible objects would correspond to an output unit in the network. For this purpose, the sigmoid activation and squared loss are replaced by softmax activation and cross-entropy loss.

The softmax is given by:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \ .$$

(m) When $K > 3$, why might sigmoid units be a bad idea?