CS 188
Summer 2023

Discussion 7A

1 Perceptron

We would like to use a perceptron to train a classifier with 2 features per point and labels +1 or −1. Consider the following labeled training data:

Features (x_1, x_2)	Label y^*
$(-1, 2)$	1
$(3, -1)$	-1
$(1, 2)$	-1
$(3, 1)$	1

1. Our two perceptron weights have been initialized to $w_1 = 2$ and $w_2 = -2$. After processing the first point with the perceptron algorithm, what will be the updated values for these weights?

2. After how many steps will the perceptron algorithm converge? Write "never" if it will never converge.

   Note: one step means processing one point. Points are processed in order and then repeated, until convergence.

Perceptron → Neural Nets

Instead of the standard perceptron algorithm, we decide to treat the perceptron as a single node neural network and update the weights using gradient descent on the loss function.

The loss function for one data point is $Loss(y, y^*) = \frac{1}{2}(y - y^*)^2$, where y^* is the training label for a given point and y is the output of our single node network for that point. We will compute a score $z = w_1x_1 + w_2x_2$, and then predict the output using an activation function g: $y = g(z)$.

1. Given a general activation function $g(z)$ and its derivative $g'(z)$, what is the derivative of the loss function with respect to w_1 in terms of $g, g', y^*, x_1, x_2, w_1$, and w_2?

   $\frac{\partial Loss}{\partial w_1} =$

2. For this question, the specific activation function that we will use is

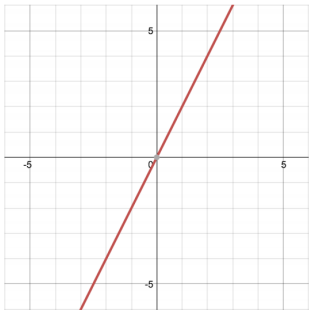
$$g(z) = 1 \text{ if } z \geq 0 \text{ , or } -1 \text{ if } z < 0$$

Given the gradient descent equation $w_i \leftarrow w_i - \alpha \frac{\partial Loss}{\partial w_1}$, update the weights for a single data point. With initial weights of $w_1 = 2$ and $w_2 = -2$, what are the updated weights after processing the first point?

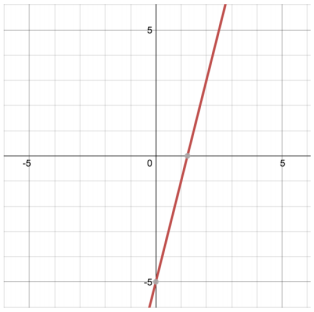
3. What is the most critical problem with this gradient descent training process with that activation function?

2 Neural Network Representations

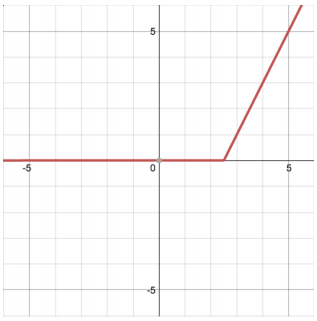
You are given a number of functions (a-h) of a single variable, x, which are graphed below. The computation graphs on the following pages will start off simple and get more complex, building up to neural networks. For each computation graph, indicate which of the functions below they are able to represent.



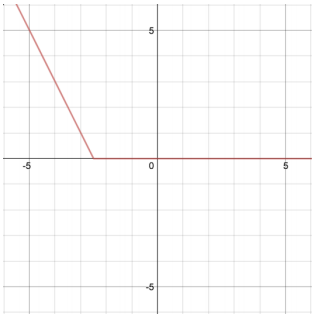
(a) $2x$



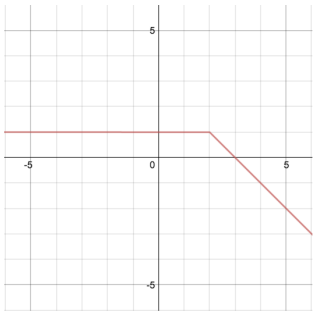
(b) $4x - 5$



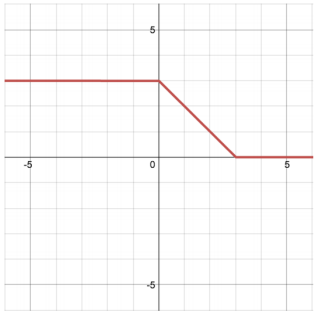
(c) $\begin{cases} 2x - 5 & x \geq 2.5 \\ 0 & x < 2.5 \end{cases}$



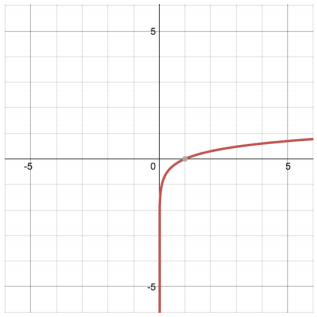
(d) $\begin{cases} -2x - 5 & x \leq -2.5 \\ 0 & x > -2.5 \end{cases}$



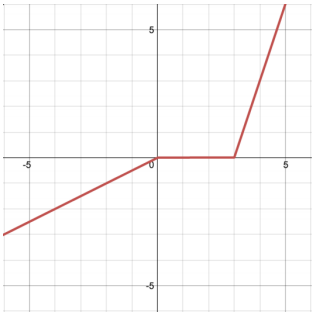
(e) $\begin{cases} -x + 3 & x \geq 2 \\ 1 & x < 2 \end{cases}$



(f) $\begin{cases} 3 & x \leq 0 \\ 3 - x & 0 < x \leq 3 \\ 0 & x > 3 \end{cases}$



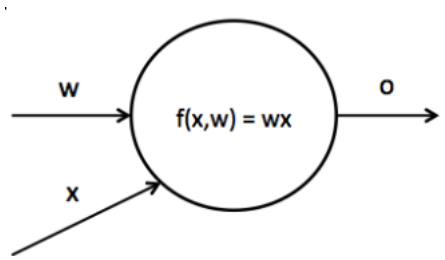
(g) $\log(x)$



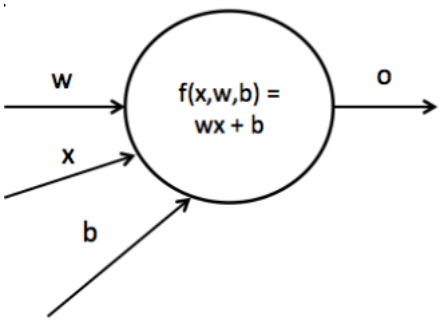
(h) $\begin{cases} 0.5x & x \leq 0 \\ 0 & 0 < x \leq 3 \\ 3x - 9 & x > 3 \end{cases}$

For each of the following computation graphs, determine which functions can be represented by the graph. In parts 1-5, write out the appropriate values of all w's and b's for each function that can be represented.

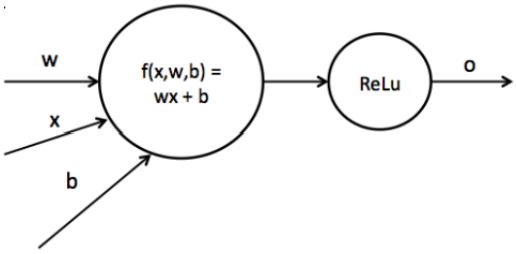
1. Linear Transformation



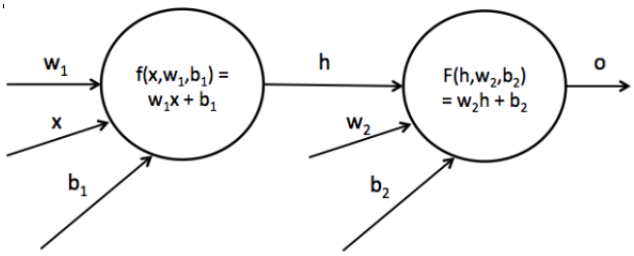
2. Linear plus Bias (aka affine transformation)



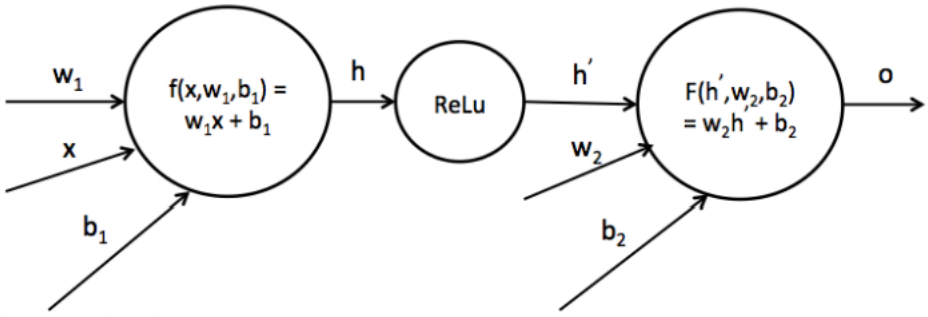
3. Nonlinearity after Linear layer



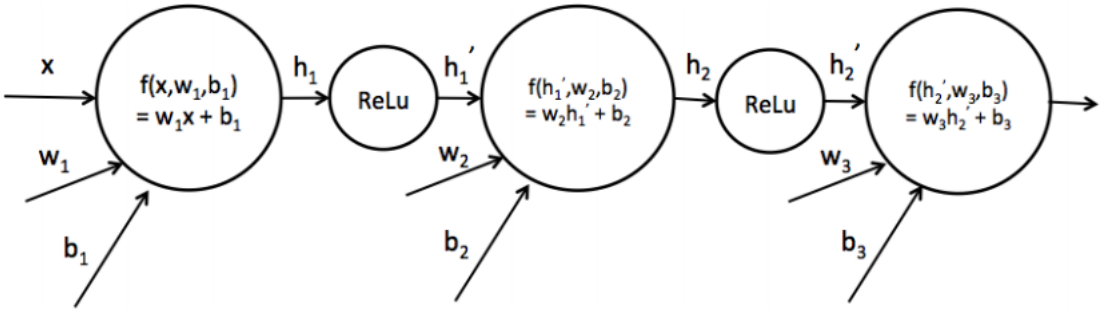
4. Composition of Affine layers



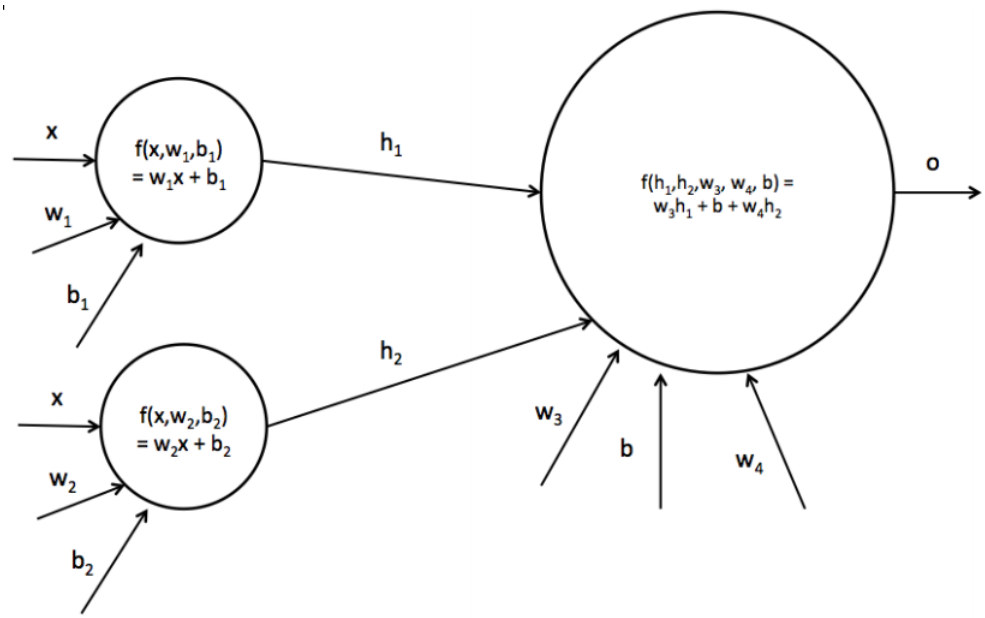
5. Two Affine layers with nonlinearity in between (hidden layer)



6. Add another hidden layer



7. Hidden layer of size 2, no nonlinearities



8. Add nonlinearities between layers