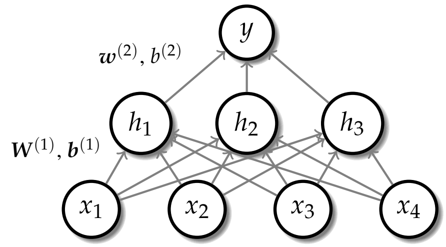
Instructor: Vishnu Boddeti
CSE 891-001: Deep Learning
Homework 1
September, 16

**Instructions:**

- **Filename:** Submit solutions in PDF format titled `written-assignment-1-{msunetid}.pdf`. Submissions in other formats or with other filenames will not be graded. You can produce your file however you like, LaTeX, Word or scan. Handwritten scans that are not legible will not be graded.
- **Submission:** Only homeworks uploaded to Google Classroom will be graded. Make sure to show all the steps of your derivations in order to receive full credit.
- **Integrity and Collaboration:** You are expected to work on the homeworks by yourself. You are not permitted to discuss them with anyone except the instructor. The homework that you hand in should be entirely your own work. You may be asked to demonstrate how you got any results that you report.
- **Clarifications:** If you have any question, please look at Google Classroom first. Other students may have encountered the same problem, and is solved already. If not, post your question there. We will respond as soon as possible.

## 1 Hard-Coding a Multilayer Perceptron (2pts):

In this problem you will find a set of weights and biases for a multilayer perceptron which determines if a list of four numbers are sorted in ascending order. More specifically, you receive 4 inputs $x_1$, $x_2$, $x_3$ and $x_4$ where $x_i \in \mathbb{R}$, and the network must output 1 if $x_1 < x_2 < x_3 < x_4$ and 0 otherwise. You will use the following Multilayer Perceptron (MLP) architecture consisting of one input layer with four nodes, one hidden layer with three nodes and one output layer with one node.



$y$
$\boldsymbol{w}^{(2)}, b^{(2)}$
$h_1$
$h_2$
$h_3$
$\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}$
$x_1$
$x_2$
$x_3$
$x_4$

The activation function of the hidden units and the output unit can be assumed to be a hard threshold function.

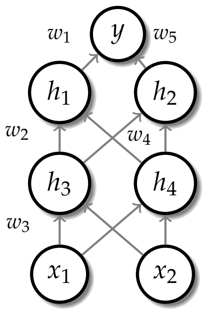
$$\phi(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

Find a set of weights and biases for the network which correctly implements this function (including cases where some of the inputs are equal). Your answer should include:

1. A $3 \times 4$ weight matrix $\boldsymbol{W}^{(1)}$ for the hidden layer.
2. A 3-dimensional vector of biases $\boldsymbol{b}^{(1)}$ for the hidden layer.
3. A 3-dimensional vector of weights $\boldsymbol{w}^{(2)}$ for the output layer.
4. A scalar bias $b^{(2)}$ for the output layer.

## 2 Sparsifying Activation Function (3pts):

An interesting property of the ReLU activation function is that it sparsifies the activations and the derivatives, i.e., sets a large fraction of the values to zeros for any given input vector. Consider the following network where the activation function of all the units is a ReLU function.



where each $w_i$ refers to the weight of a single connection. Assume we are trying to minimize a loss function $\mathcal{L}$ which depends only on the activation of the output unit $y$. Suppose the unit $h_1$ receives an input -1 on a particular training case, so the ReLU evaluates to 0. Based only on this information, which of the weight derivatives $\frac{\partial \mathcal{L}}{\partial w_1}$, $\frac{\partial \mathcal{L}}{\partial w_2}$, $\frac{\partial \mathcal{L}}{\partial w_3}$ are guaranteed to be 0 for this training case? Provide a YES or NO answer for each with your justification.

## 3 Universal Approximation Theorem (5pts):

In this problem you will build the intuition behind how the neural network function class can approximate a particular class of functions arbitrarily well.

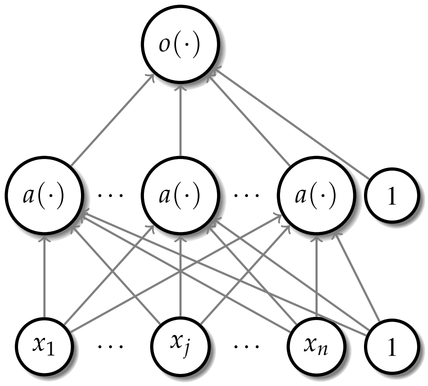
Suppose $f : I \to \mathbb{R}$, where $I = [a, b] \subset \mathbb{R}$ and $a \leq b$ is a closed interval. Also, let $\hat{f}_\tau : I \to \mathbb{R}$ be some function approximator from our network where $\tau$ is a description of our network architecture and weights. Here, $\tau$ is a tuple of $(n, \boldsymbol{W}_0 \in \mathbb{R}^{n\times 1}, \boldsymbol{b}_0 \in \mathbb{R}^n, \boldsymbol{W}_1 \in \mathbb{R}^{n\times 1}, \boldsymbol{b}_1 \in \mathbb{R}^n)$, where $n$ is the hidden layer size, $\boldsymbol{W}_0$ and $\boldsymbol{b}_0$ describe the input hidden parameters, and $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ describe the output hidden parameters.

The output is computed as $\hat{f}_\tau(\boldsymbol{x}) = \boldsymbol{W}_1 a(\boldsymbol{W}_0\boldsymbol{x} + \boldsymbol{b}_0) + \boldsymbol{b}_1$, where the activation $a(\cdot)$ is an indicator function, i.e., $a(y) = \mathbb{I}(y \geq 0)$, where $\mathbb{I}(s)$ is 1 when the boolean value $s$ is true and 0 otherwise. For a vector, the activation function is applied to each element of the vector.

We want to show that there exist a series of neural networks $\{\tau_i\}_{i=1}^N$ such that:

$$\forall \epsilon > 0, \exists M : \forall m > M, \|f - \hat{f}_{\tau_m}\} < \epsilon$$

where $\|f - \hat{f}\| = \int_I |f(x) - \hat{f}(x)|dx$.


$o(\cdot)$
$a(\cdot)$ $\cdots$ $a(\cdot)$ $\cdots$ $a(\cdot)$ 1
$x_1$ $\cdots$ $x_j$ $\cdots$ $x_n$ 1

1. **(1.0 pt)** Consider a rectangular function $g(h, a, b, x) = h \cdot \mathbb{I}(a \leq x \leq b)$. Given some $(h, a, b)$ show $\exists\tau : \hat{f}_\tau(x) = g(h, a, b, x)$. You answer should be a specific choice of $n$, $\boldsymbol{W}_0$, $\boldsymbol{b}_0$, $\boldsymbol{W}_1$, and $\boldsymbol{b}_1$, which will be functions of the selected $(h, a, b)$, where $h \in \mathbb{R}$, $a \in \mathbb{R}$, and $b \in \mathbb{R}$.

2. **(1.5 pt)** Given $f(x) = -x^2 + 1$ where $I = [-1, 1]$ and some initial function $\hat{f}_0(x) = 0$ which is identically 0, construct a new function $\hat{f}_1(x) = \hat{f}_0(x) + g(h_1, a_1, b_1, x)$ such that $\|f - \hat{f}_1\| \leq \|f - \hat{f}_0\|$, with the rectangle function in the previous question. Note that $h_1$, $a_1$, and $b_1$ are going to depend on your choice of $f$, $\hat{f}$ and $I$. Plot $f$ and $\hat{f}_1$, write down $h_1$, $a_1$, and $b_1$, and justify why $\|f - \hat{f}_1\| < \|f - \hat{f}_0\|$.

3. **(2.5 pt)** Describe a procedure which starts with $\hat{f}_0(x) = 0$ and a fixed $N$, then construct a series $\{\hat{f}_i\}_{i=0}^N$ where $\hat{f}_{i+1}(x) = \hat{f}_i(x) + g(h_{i+1}, a_{i+1}, b_{i+1}, x)$, which satisfies $\|f - \hat{f}_{i+1}\| < \|f - \hat{f}_i\|$. Use the definition of $g$ from above and the choice of $f$ from the previous question. Plot $f$, $\hat{f}_1$,$\hat{f}_2$ and $\hat{f}_3$, write down how to generate $h_{i+1}$,$a_{i+1}$,$b_{i+1}$, and justify why $\|f - \hat{f}_{i+1}\| < \|f - \hat{f}_i\|$.