Due Date : February 4th (11pm), 2020

Instructions

- *For all questions, show your work!*
- *Use LaTeX and the template we provide when writing your answers. You may reuse most of the notation shorthands, equations and/or tables. See the assignment policy on the course website for more details.*
- *Submit your answers electronically via Gradescope.*
- ***TAs for this assignment are Jie Fu, Sai Rajeswar, and Akilesh B***

Question 1 (4-4-4)**.** Using the following definition of the derivative and the definition of the Heaviside step function :

$$\frac{d}{dx}f(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon} \qquad H(x) = \begin{cases} 1 & \text{if } x > 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

1. Show that the derivative of the rectified linear unit $g(x) = \max\{0, x\}$, **wherever it exists**, is equal to the Heaviside step function.
2. Give two alternative definitions of $g(x)$ using $H(x)$.
3. Show that $H(x)$ can be well approximated by the sigmoid function $\sigma(x) = \frac{1}{1+e^{-kx}}$ asymptotically (i.e for large k), where k is a parameter.

Answer 1.

1.

$$\frac{d}{dx}g(x) = \lim_{\epsilon \to 0} \frac{g(x+\epsilon) - g(x)}{\epsilon}$$

$$\frac{d}{dx}g(x) = \lim_{\epsilon \to 0} \frac{max(0, x+\epsilon) - max(0, x)}{\epsilon}$$

For x>0,

$$\frac{d}{dx}g(x) = \lim_{\epsilon \to 0} \frac{(x+\epsilon) - (x)}{\epsilon} = 1$$

For x<0,

$$\frac{d}{dx}g(x) = \lim_{\epsilon \to 0} \frac{(0) - (0)}{\epsilon} = 0$$

For x=0,

$$\frac{d}{dx}g(x) = \lim_{\epsilon \to 0^-} \frac{(0) - (0)}{\epsilon} = 0$$

$$\frac{d}{dx}g(x) = \lim_{\epsilon \to 0^+} \frac{(\epsilon) - (0)}{\epsilon} = 1$$

Since the left and right handed limit is not equal for x=0, we say that the derivative of g(x) is undefined. So we can say that derivative of g(x) is equal to heaviside step function wherever it exists

$$g^{'}(x) = \begin{cases} 1 & \text{if } x > 0 \\ Undefined & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

2. Two alternate definitions of g(x) using H(x)
   (a) g(x) = x. H(x)
   (b) g(x) = x. (1 - H(-x))
3. For asymptotically large k value,

$$\sigma(x) = \frac{1}{1+e^{-kx}}$$

For x>0,

$$\lim_{k\to\infty} \frac{1}{1+e^{-kx}} = 1 \quad \textit{As } e^{-kx} \textit{ approaches to 0}$$

For x<0,

$$\lim_{k\to\infty} \frac{1}{1+e^{-kx}} = 0 \quad \textit{As } e^{kx} \textit{ approaches to } \infty$$

For x=0,

$$\lim_{k\to\infty} \frac{1}{1+e^{-kx}} = \frac{1}{2} \quad \textit{As } e^{0} \textit{ is equal to 1}$$

$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0, k \to \infty \\ \frac{1}{2} & \text{if } x = 0, k \to \infty \\ 0 & \text{if } x < 0, k \to \infty \end{cases} \qquad H(x) = \begin{cases} 1 & \text{if } x > 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Therefore, we can say that H(x) can be approximated by the sigmoid function $\sigma(x) = \frac{1}{1+e^{-kx}}$ asymptotically (i.e for large k)

Question 2 (3-3-3-3)**.** Recall the definition of the softmax function : $S(\boldsymbol{x})_i = e^{\boldsymbol{x}_i}/\sum_j e^{\boldsymbol{x}_j}$.

1. Show that softmax is translation-invariant, that is : $S(\boldsymbol{x}+c) = S(\boldsymbol{x})$, where c is a scalar constant.
2. Show that softmax is not invariant under scalar multiplication. Let $S_c(\boldsymbol{x}) = S(c\boldsymbol{x})$ where $c \geq 0$. What are the effects of taking c to be 0 and arbitrarily large ?
3. Let $\boldsymbol{x}$ be a 2-dimensional vector. One can represent a 2-class categorical probability using softmax $S(\boldsymbol{x})$. Show that $S(\boldsymbol{x})$ can be reparameterized using sigmoid function, i.e. $S(\boldsymbol{x}) = [\sigma(z), 1-\sigma(z)]^\top$ where z is a scalar function of $\boldsymbol{x}$.
4. Let $\boldsymbol{x}$ be a K-dimensional vector ($K \geq 2$). Show that $S(\boldsymbol{x})$ can be represented using $K-1$ parameters, i.e. $S(\boldsymbol{x}) = S([0, y_1, y_2, ..., y_{K-1}]^\top)$ where y_i is a scalar function of $\boldsymbol{x}$ for $i \in \{1, ..., K-1\}$.

Answer 2.

1.

$$\begin{aligned} S(\boldsymbol{x})_i &= \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}} \\ S(\boldsymbol{x}+c)_i &= \frac{e^{\boldsymbol{x}_i+c}}{\sum_j e^{\boldsymbol{x}_j+c}} = \frac{e^{\boldsymbol{x}_i}.e^{c}}{\sum_j e^{\boldsymbol{x}_j}.e^{c}} = \frac{e^{c}.e^{\boldsymbol{x}_i}}{e^{c}.\sum_j e^{\boldsymbol{x}_j}} = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}} \\ S(\boldsymbol{x}+c) &= S(\boldsymbol{x})_i \end{aligned}$$

Hence, we can say that softmax is translation-invariant.

2.

$$S(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$$

$$S(c.\boldsymbol{x}_i) = \frac{e^{c.\boldsymbol{x}_i}}{\sum_j e^{c.\boldsymbol{x}_j}} \neq \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$$
$$S(c.\boldsymbol{x}) \neq S(\boldsymbol{x})$$

For c = 0,

$$S(c.\boldsymbol{x}_i) = \frac{e^{c.\boldsymbol{x}_i}}{\sum_j e^{c.\boldsymbol{x}_j}} = \frac{1}{N} \quad \text{where N is the number of elements in x}$$

For $c \to \infty$,

$$\lim_{c\to\infty} S(c.\boldsymbol{x}_i) = \frac{e^{c.\boldsymbol{x}_i}}{\sum_j e^{c.\boldsymbol{x}_j}} = 1 \text{ for maximum value of x and 0 elsewhere.}$$

Hence we can say that softmax is not invariant under scalar multiplication.

3.

$$S(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$$

For two class problem.

$$S(\boldsymbol{x})_1 = \frac{e^{\boldsymbol{x}_1}}{e^{\boldsymbol{x}_1} + e^{\boldsymbol{x}_2}} \qquad S(\boldsymbol{x})_2 = 1 - S(\boldsymbol{x})_1$$

$$S(\boldsymbol{x})_1 = \frac{1}{1 + e^{\boldsymbol{x}_2 - \boldsymbol{x}_1}}$$
$$S(\boldsymbol{x})_1 = \frac{1}{1 + e^{-(\boldsymbol{x}_1 - \boldsymbol{x}_2)}}$$
$$S(\boldsymbol{x})_1 = \sigma(\boldsymbol{x}_1 - \boldsymbol{x}_2)$$
$$S(\boldsymbol{x})_2 = 1 - \sigma(\boldsymbol{x}_1 - \boldsymbol{x}_2)$$

For 2 dimensional vector x, and z = $(x_1 - x_2)$, Softmax function $S(\boldsymbol{x})$ can be reparameterized using sigmoid function $[\sigma(z), 1 - \sigma(z)]^\top$

4.

$$S(x)_1 = \frac{e^{x_1}}{e^{x_1} + e^{x_2} + .. + s^{x_k}}$$

Multiply and divide the above equation by e^{-x_1}

$$S(x)_1 = \frac{e^0}{e^0 + e^{x_2 - x_1} + .. + s^{x_k - x_1}}$$

$$S(x)_2 = \frac{e^{x_2 - x_1}}{e^0 + e^{x_2 - x_1} + .. + s^{x_k - x_1}}$$

$$S(x)_k = \frac{e^{x_k - x_1}}{e^0 + e^{x_2 - x_1} + .. + s^{x_k - x_1}}$$

So we can say that $S(\boldsymbol{x}) = S([0, y_1, y_2, ..., y_{K-1}]^\top)$ where $y_i = x_{i+1} - x_1$ for $i \in \{1, ..., K-1\}$.

Question 3 (16)**.** Consider a 2-layer neural network $y : \mathbb{R}^D \to \mathbb{R}^K$ of the form :

$$y(x, \Theta, \sigma)_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} \sigma \left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \right) + \omega_{k0}^{(2)}$$

for $1 \leq k \leq K$, with parameters $\Theta = (\omega^{(1)}, \omega^{(2)})$ and logistic sigmoid activation function σ. Show that there exists an equivalent network of the same form, with parameters $\Theta' = (\tilde{\omega}^{(1)}, \tilde{\omega}^{(2)})$ and tanh activation function, such that $y(x, \Theta', \tanh) = y(x, \Theta, \sigma)$ for all $x \in \mathbb{R}^D$, and express Θ' as a function of Θ.

Answer 3. We can express sigmoid activation function in the form of tanh function.

$$\begin{aligned}
\tanh(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}} \\
\tanh(\frac{x}{2}) &= \frac{e^{\frac{x}{2}} - e^{-\frac{x}{2}}}{e^{\frac{x}{2}} + e^{-\frac{x}{2}}} \\
\tanh(\frac{x}{2}) + 1 &= \frac{e^{\frac{x}{2}} - e^{-\frac{x}{2}}}{e^{\frac{x}{2}} + e^{-\frac{x}{2}}} + 1 = \frac{2e^{\frac{x}{2}}}{e^{\frac{x}{2}} + e^{-\frac{x}{2}}} \\
\frac{1}{2}(\tanh(\frac{x}{2}) + 1) &= \frac{e^{\frac{x}{2}}}{e^{\frac{x}{2}} + e^{-\frac{x}{2}}} = \frac{1}{1 + \frac{e^{-\frac{x}{2}}}{e^{\frac{x}{2}}}} = \frac{1}{1 + e^{-x}} = \sigma(x) \\
\therefore \sigma(x) &= \frac{1}{2}(\tanh(\frac{x}{2}) + 1)
\end{aligned}$$

Replacing sigmoid with tanh equivalent function in the above given equation.

$$y(x, \Theta, \sigma)_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} \sigma \left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \right) + \omega_{k0}^{(2)}$$

$$y(x, \Theta, \tanh)_k = \sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} \left(\tanh \left(\sum_{i=1}^{D} \frac{\omega_{ji}^{(1)}}{2} x_i + \frac{\omega_{j0}^{(1)}}{2} \right) + 1 \right) + \omega_{k0}^{(2)}$$

$$y(x, \Theta, \tanh)_k = \sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} \tanh \left(\sum_{i=1}^{D} \frac{\omega_{ji}^{(1)}}{2} x_i + \frac{\omega_{j0}^{(1)}}{2} \right) + \sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} + \omega_{k0}^{(2)}$$

$$y(x, \Theta', \tanh)_k = \sum_{j=1}^{M} \tilde{\omega}_{kj}^{(2)} \sigma \left(\sum_{i=1}^{D} \tilde{\omega}_{ji}^{(1)} x_i + \tilde{\omega}_{j0}^{(1)} \right) + \tilde{\omega}_{k0}^{(2)}$$

The equivalent network with Θ' in the form of Θ such that $y(x, \Theta', \tanh) = y(x, \Theta, \sigma)$

$$\tilde{\omega}_{kj}^{(2)} = \frac{\omega_{kj}^{(2)}}{2} \qquad \tilde{\omega}_{k0}^{(2)} = \sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} + \omega_{k0}^{(2)}$$

$$\tilde{\omega}_{ji}^{(1)} = \frac{\omega_{ji}^{(1)}}{2} \qquad \tilde{\omega}_{j0}^{(1)} = \frac{\omega_{j0}^{(1)}}{2}$$

TABLE 1 – Forward AD, with $y = f(x_1, x_2) = 1/(x_1 + x_2) + {x_2}^2 + \cos(x_1)$ at $(x_1, x_2) = (3, 6)$ and setting $\dot{x}_1 = 1$ to compute $\partial y/\partial x_1$.

	Forward evaluation trace				Forward derivative trace		
	v_{-1}	$= x_1$	$= 3$		$= \dot{v}_{-1}$	$\dot{x}_1$	$= 1$
	v_0	$= x_2$	$= 6$		$= \dot{v}_0$	$\dot{x}_2$	$= 0$
	v_1	$= v_{-1} + v_0$	$= 3 + 6$		$\dot{v}_1$	$= \dot{v}_{-1} + \dot{v}_0$	$= 1 + 0$
	v_2	$= 1/v_1$	$= 0.1111 \times 5$		$\dot{v}_2$	$= \dot{v}_{-1}/(v_1)^2$	$= -1/9^2$
$\Downarrow$	v_3	$= (v_0)^2$	$= 6 \times 6$	$\Downarrow$	$\dot{v}_3$	$= 2 \times v_0 \times \dot{v}_0$	$= 2 \times 6 \times 0$
	v_4	$= \cos v_{-1}$	$= -0.9899$		$\dot{v}_4$	$= -\sin v_{-1} \times \dot{v}_{-1}$	$= -0.1411 \times 1$
	v_5	$= v_2 + v_3$	$= 36 + 0.1111$		$\dot{v}_5$	$= \dot{v}_2 + \dot{v}_3$	$= -0.0123 + 0$
	v_6	$= v_4 + v_5$	$= -0.9899 + 36.1111$		$\dot{v}_6$	$= \dot{v}_4 + \dot{v}_5$	$= -0.012 - 0.141$
	y	$= v_6$	$= 35.1212$		$= \dot{y}$	$\dot{v}_6$	$= -0.1534$

TABLE 2 – Reverse AD, with $y = f(x_1, x_2) = 1/(x_1 + x_2) + {x_2}^2 + \cos(x_1)$ at $(x_1, x_2) = (3, 6)$. Setting $\bar{y} = 1$, $\partial y/\partial x_1$ and $\partial y/\partial x_2$ can be computed together.

	Forward evaluation trace		
	v_{-1}	$= x_1$	$= 3$
	v_0	$= x_2$	$= 6$
	v_1	$= v_{-1} + v_0$	$= 3 + 6$
	v_2	$= 1/v_1$	$= 0.1111 \times 5$
$\Downarrow$	v_3	$= (v_0)^2$	$= 6 \times 6$
	v_4	$= \cos v_{-1}$	$= -0.9899$
	v_5	$= v_2 + v_3$	$= 36 + 0.1111$
	v_6	$= v_4 + v_5$	$= -0.9899 + 36.1111$
	y	$= v_6$	$= 35.1212$

	Reverse adjoint trace		
	$\bar{x}_1$	$= \bar{v}_{-1}$	$= -0.1534$
	$\bar{x}_2$	$= \bar{v}_0$	$= 11.9876$
	$\bar{v}_0$	$= \bar{v}_0 + \bar{v}_1 \frac{\partial v_1}{\partial v_0}$	$= 11.9876$
	$\bar{v}_{-1}$	$= \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= -0.1534$
$\Uparrow$	$\bar{v}_1$	$= \bar{v}_2 \frac{\partial v_2}{\partial v_1}$	$= -0.0123$
	$\bar{v}_0$	$= \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= 12$
	$\bar{v}_{-1}$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_{-1}}$	$= -0.1411$
	$\bar{v}_2$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_2}$	$= 1$
	$\bar{v}_3$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= 1$
	$\bar{v}_4$	$= \bar{v}_6 \frac{\partial v_6}{\partial v_4}$	$= 1$
	$\bar{v}_5$	$= \bar{v}_6 \frac{\partial v_6}{\partial v_5}$	$= 1$
	$\bar{v}_6$	$= \bar{y}$	$= 1$

Question 4 (5-5)**.** Fundamentally, back-propagation is just a special case of reverse-mode Automatic Differentiation (AD), applied to a neural network. Based on the "three-part" notation shown in Table 1 and 2, represent the evaluation trace and derivative (adjoint) trace of the following examples. In the last columns of your solution, numerically evaluate the value up to 4 decimal places.

1. Forward AD, with $y = f(x_1, x_2) = 1/(x_1 + x_2) + {x_2}^2 + \cos(x_1)$ at $(x_1, x_2) = (3, 6)$ and setting $\dot{x}_1 = 1$ to compute $\partial y/\partial x_1$.
2. Reverse AD, with $y = f(x_1, x_2) = 1/(x_1 + x_2) + {x_2}^2 + \cos(x_1)$ at $(x_1, x_2) = (3, 6)$. Setting $\bar{y} = 1$, $\partial y/\partial x_1$ and $\partial y/\partial x_2$ can be computed together.

Answer 4. Answers in the above two tables

1. Table 1
2. Table 2

Question 5 (6)**.** Compute the *full*, *valid*, and *same* convolution (with kernel flipping) for the following 1D matrices : $\begin{bmatrix}1,2,3,4\end{bmatrix} * \begin{bmatrix}1,0,2\end{bmatrix}$

Answer 5. Kernel = [1,0,2]
Flipped Kernel = [2,0,1]
Matrix = [0,1,2,3,4,0]

Full : Padding with 2 zero each side.
Valid : No Padding
Same : (Half Convolution) Padding such that the output dimension is same as input dimension. So we pad 1 zero in the start and end.

Full : $\begin{bmatrix}1,2,5,8,6,8\end{bmatrix}$; Valid : $\begin{bmatrix}5,8\end{bmatrix}$; Same : $\begin{bmatrix}2,5,8,6\end{bmatrix}$.

Question 6 (5-5)**.** Consider a convolutional neural network. Assume the input is a colorful image of size 256×256 in the RGB representation. The first layer convolves 64 8×8 kernels with the input, using a stride of 2 and no padding. The second layer downsamples the output of the first layer with a 5×5 non-overlapping max pooling. The third layer convolves 128 4×4 kernels with a stride of 1 and a zero-padding of size 1 on each border.

1. What is the dimensionality (scalar) of the output of the last layer ?
2. Not including the biases, how many parameters are needed for the last layer ?

Answer 6.
First Layer
Input Matrix = 256 X 256 X 3 Kernel Matrix = 8 X 8 X 3 Num of Kernels = 64 Padding = 0 Stride = 2

Output Dimension = ((256 - 8 + 2(0))/2) + 1 = 125 = 125 X 125 X 64

Second Layer
Input Matrix = 125 X 125 X 64 Pooling Layer on this output : Pooling Layer Dimension : 5 X 5 with no overlapping. Stride = 5

Output Dimension = ((125 - 5 + 2(0))/5) + 1 = 25 = 25 X 25 X 64

Third Layer
Input Matrix = 125 X 125 X 64 Kernel Matrix = 4 X 4 X 64 Num of Kernels = 128 Padding = 1 Stride = 1

Output Dimension = ((25 - 4 + 2(1))/1) + 1 = 24 = 24 X 24 X 128

1. Dimensionality(scalar) of output of last layer : 24X24X128 = **73728**
2. Parameters for last layer = 128 X (Kernel dimension) = 128 X (4X4X64) = **131072**

Question 7 (4-4-6)**.** Assume we are given data of size $3 \times 64 \times 64$. In what follows, provide a correct configuration of a convolutional neural network layer that satisfies the specified assumption. Answer with the window size of kernel (k), stride (s), padding (p), and dilation (d, with convention $d = 1$ for no dilation). Use square windows only (e.g. same k for both width and height).

1. The output shape (o) of the first layer is $(64, 32, 32)$.
   (a) Assume $k = 8$ without dilation.
   (b) Assume $d = 7$, and $s = 2$.
2. The output shape of the second layer is $(64, 8, 8)$. Assume $p = 0$ and $d = 1$.
   (a) Specify k and s for pooling with non-overlapping window.
   (b) What is output shape if $k = 8$ and $s = 4$ instead ?
3. The output shape of the last layer is $(128, 4, 4)$.
   (a) Assume we are not using padding or dilation.
   (b) Assume $d = 2$, $p = 2$.
   (c) Assume $p = 1$, $d = 1$.

Answer 7. Fill up the following table,

$$o = \lfloor \frac{i + 2p - k}{s} \rfloor + 1$$

In case of dilations,

$$\text{effective k'} = k + (k-1)(d-1)$$

There are cases where multiple combinations are possible but since nothing is conveyed, the valid values are chosen arbitrarily.

		i	p	d	k	s	o
1.	(a)	64	3	1	8	2	32
	(b)	64	3	7	2	2	32
2.	(a)	32	0	1	4	4	8
	(b)	32	0	1	8	4	7
3.	(a)	8	0	1	2	2	4
	(b)	8	2	2	2	3	4
	(c)	8	1	1	4	2	4