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.

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

- if x $>$ 0 :

$$f(x) = \max\{0, x\} = x, H(x) = 1$$

Hence,

$$\frac{d}{dx}f(x) = \lim_{\epsilon \to 0} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon}$$
$$\frac{d}{dx}f(x) = \lim_{\epsilon \to 0} \frac{x+\epsilon - x}{\epsilon} = 1 = H(x)$$

- if x $<$ 0 :

$$f(x) = \max\{0, x\} = 0, H(x) = 0$$

Hence,

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

- if x = 0 : Hence,

$$\lim_{\epsilon\to 0^-} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon} = \lim_{\epsilon\to 0^-} \frac{0-0}{\epsilon} = 0$$
$$\lim_{\epsilon\to 0^+} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon} = \lim_{\epsilon\to 0|} \frac{\epsilon-0}{\epsilon} = 1$$
$$\lim_{\epsilon\to 0^-} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon} \neq \lim_{\epsilon\to 0^+} \frac{\max\{0, x+\epsilon\} - \max\{0, x\}}{\epsilon}$$

Hence, derivative of $f(x) = \max\{0, x\}$ doesn't exist at $x = 0$. For other cases, $\frac{d}{dx}f(x) = H(x)$.

2. Below are the two alternate definition of $g(x)$
   - $g(x) = xH(x)$
   - $g(x) = x(1 - H(-x))$
3. Here,
   - if x > 0 : H(x) = 1

$$\frac{d}{dx}\sigma(x) = \lim_{k\to\infty} \frac{1}{1+e^{-kx}} = \frac{1}{1+0} = 1 = H(x)$$

   - if x = 0 : H(x) = $\frac{1}{2}$

$$\frac{d}{dx}\sigma(x) = \lim_{k\to\infty} \frac{1}{1+e^{-kx}} = \frac{1}{1+e^{0}} = \frac{1}{1+1} = \frac{1}{2} = H(x)$$

   - if x < 0 : H(x) = 0

$$\frac{d}{dx}\sigma(x) = \lim_{k\to\infty} \frac{1}{1+e^{-kx}} = \frac{1}{1+\infty} = 0 = H(x)$$

   From above, we show that $H(x)$ can be well approximated with a Sigmoid function asymptotically.

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. $S(\boldsymbol{x}+c) = (S(\boldsymbol{x}+c)_1, S(\boldsymbol{x}+c)_2, ..., S(\boldsymbol{x}+c)_K)$
   $S(\boldsymbol{x}+c)_i = \frac{e^{\boldsymbol{x}_i+c}}{\sum_j e^{\boldsymbol{x}_j+c}} = \frac{e^c e^{\boldsymbol{x}_i}}{e^c \sum_j e^{\boldsymbol{x}_j}} = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}} = S(\boldsymbol{x})_i$
   For an arbitrary constant c, we proved that $S(\boldsymbol{x}+c)_i = S(\boldsymbol{x})_i$. Hence, $S(\boldsymbol{x}+c) = S(\boldsymbol{x})$.

2. $S(c\boldsymbol{x})_i = \frac{e^{c\boldsymbol{x}_i}}{\sum_j e^{c\boldsymbol{x}_j}} \neq \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$
   For an arbitrary constant c, we proved that $S(c\boldsymbol{x})_i \neq S(\boldsymbol{x})_i$. Hence, $S(c\boldsymbol{x}) \neq S(\boldsymbol{x})$.
   - if $c = 0$
     , $S(c\boldsymbol{x})_i = \frac{e^{c\boldsymbol{x}_i}}{\sum_j e^{c\boldsymbol{x}_j}} = \frac{e^0}{\sum_j e^0} = \frac{1}{\sum_j 1} = \frac{1}{K}$
     Hence, if $c = 0$, our Softmax act as uniform distribution over the input.
   - if $c \to \infty$,
     — if $x_i = x_j = a \quad \forall i, j \in \{1, 2, ..., K\}$,
     $S(c\boldsymbol{x}) = \frac{e^{c\boldsymbol{x}_i}}{\sum_j e^{c\boldsymbol{x}_j}} = \frac{e^{ca}}{\sum_j e^{ca}} = \frac{1}{K}$
     Hence, if all values of $\boldsymbol{x}$ are same, we will end up with a uniform distribution.
     — if $\boldsymbol{x}$ has distinct values, $S(c\boldsymbol{x}) \to [0, 0, ..., 1, ..0]$. It will be more like a one-hot representation. The maximum value of $\boldsymbol{x}$ will get close to 1 and rest values will go towards 0.
     — if $\boldsymbol{x}$ has multiple max values, their corresponding softmax value will be similar. Let's say the max value occurs r times in $\boldsymbol{x}$, then for those max values, their softmax value will be $\frac{1}{r}$. While the softmax values of the rest of the entries will go towards 0.
3. $\boldsymbol{x} = [x_1, x_2]$
   $S(\boldsymbol{x}) = [\frac{e^{x_1}}{e^{x_1}+e^{x_2}}, \frac{e^{x_2}}{e^{x_1}+e^{x_2}}] = [\frac{e^{x_1}}{e^{x_1}+e^{x_2}}, 1 - \frac{e^{x_1}}{e^{x_1}+e^{x_2}}] = [\frac{1}{1+e^{-(x_1-x_2)}}, 1 - \frac{1}{1+e^{-(x_1-x_2)}}]$
   $S(\boldsymbol{x}) = [\sigma(x_1 - x_2), 1 - \sigma(x_1 - x_2)] = [\sigma(z), 1 - \sigma(z)] \quad$ Here, $z = x_1 - x_2$
4. $S(\boldsymbol{x}) = S([x_1, x_2, x_3, ..., x_K]^\top)$
   Here, we know that softmax is translation invariant(from Que-2(1)). Hence, subtracting x_1 from $\boldsymbol{x}$, we get the following,
   $S(\boldsymbol{x}) = S([0, x_2 - x_1, x_3 - x_1, ..., x_K - x_1]^\top) = S([0, y_1, y_2, ..., y_{K-1}]^\top)$,
   Here $y_j = x_{j+1} - x_1$ and $j \in \{1, .., K-1\}$. Hence, we can see that $S(\boldsymbol{x})$ can be represented with $K - 1$ parameters.

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. First, let's establish relation between $tanh(x)$ and $\sigma(x)$.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$tanh(x) + 1 = \frac{2e^{2x}}{e^{2x} + 1} = \frac{2}{1 + e^{-2x}} = 2\sigma(2x)$$

From above,

$$\sigma(x) = \frac{tanh(\frac{x}{2}) + 1}{2} \quad \text{(We will use this relation for our purpose.)}$$

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

From above, we can easily express Θ' as a function of Θ as below,

$$
\tilde{\omega}_{kj}^{(2)} = \frac{\omega_{kj}^{(2)}}{2}, \quad \tilde{\omega}_{kj}^{(1)} = \frac{\omega_{kj}^{(1)}}{2} \quad \forall \quad j \in \{1, 2, ..., M\}
$$

$$
\text{and,} \quad \tilde{\omega}_{j0}^{(1)} = \frac{\omega_{j0}^{(1)}}{2}, \quad \tilde{\omega}_{k0}^{(2)} = \sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} + \omega_{k0}^{(2)}
$$

That's how we can construct equivalent network for sigmoid and tanh activations.

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 2 and **??**, 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.

Answer 4.

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		
	v_{-1}	$= x_1$	$= 3$
	v_0	$= x_2$	$= 6$
	v_1	$= v_{-1} + v_0$	$= 9$
	v_2	$= \frac{1}{v_1}$	$= \frac{1}{9}$
$\Downarrow$	v_3	$= v_0^2$	36
	v_4	$= cos(v_{-1})$	$= cos(3)$
	v_5	$= v_2 + v_3 + v_4$	$= 35.12111$
	y	$= v_5$	$= 35.12111$

	Forward derivative trace		
	$= \dot{v}_{-1}$	$\dot{x}_1$	$= 1$
	$= \dot{v}_0$	$\dot{x}_2$	$= 0$
	$\dot{v}_1$	$= \dot{v}_{-1} + \dot{v}_0$	$= 1 + 0$
	$\dot{v}_2$	$= \frac{-\dot{v}_1}{v_1^2}$	$= \frac{-1}{81}$
$\Downarrow$	$\dot{v}_3$	$= 2 v_0 \dot{v}_0$	$= 0$
	$\dot{v}_4$	$= -sin(v_{-1})\dot{v}_{-1}$	$= -0.14112$
	$\dot{v}_5$	$= \dot{v}_2 + \dot{v}_3 + \dot{v}_4$	$= -0.15346$
	$= \dot{y}$	$\dot{v}_5$	$= -0.15346$

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	$= v_{-1} + v_0$	$= 9$
	v_2	$= \frac{1}{v_1}$	$= \frac{1}{9}$
$\Downarrow$	v_3	$= v_0^2$	36
	v_4	$= cos(v_{-1})$	$= cos(3)$
	v_5	$= v_2 + v_3 + v_4$	$= 35.12111$
	y	$= v_5$	$= 35.12111$

	Reverse adjoint trace		
	$\bar{x}_1$	$= \bar{v}_{-1}$	=-0.15346
	$\bar{x}_2$	$= \bar{v}_0$	$= 11.98765$
	$\bar{v}_0$	$= \bar{v}_0 + \bar{v}_1 \frac{\partial v_1}{\partial v_0}$	$= 11.98765$
	$\bar{v}_{-1}$	$= \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= -0.15346$
$\Uparrow$	$\bar{v}_1$	$= \bar{v}_2 \frac{\partial v_2}{\partial v_1}$	$= \frac{-1}{81}$
	$\bar{v}_0$	$= \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= 12$
	$\bar{v}_{-1}$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_{-1}}$	$= -sin(3)$
	$\bar{v}_4$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= 1$
	$\bar{v}_3$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= 1$
	$\bar{v}_2$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_2}$	$= 1$
	$\bar{v}_5$	$= \bar{y}$	$= 1$

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. 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.

1. The output dimension of the last layer is 24.
2. For the last layer, we need 131072 parameters.

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,

		i	p	d	k	s	o
1.	(a)	64	3	1	8	2	32
	(b)	64	7	7	3	2	32
2.	(a)	32	0	1	3	4	8
	(b)	32	0	1	8	4	7
3.	(a)	8	0	1	5	1	4
	(b)	8	2	2	5	1	4
	(c)	8	1	1	3	2	4