

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 :

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

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.

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

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

Answer 2.

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

TABLE 1 – Forward AD example, with $y = f(x_1, x_2) = \ln(x_1) + x_1x_2 - \sin(x_2)$ at $(x_1, x_2) = (2, 5)$ and setting $\dot{x}_1 = 1$ to compute $\partial y / \partial x_1$.

Forward evaluation trace			Forward derivative trace		
v_{-1}	$= x_1$	$= 2$	$= \dot{v}_{-1}$	\dot{x}_1	$= 1$
v_0	$= x_2$	$= 5$	$= \dot{v}_0$	\dot{x}_2	$= 0$
v_1	$= \ln(v_{-1})$	$= \ln(2)$	\dot{v}_1	$= \dot{v}_{-1} / v_{-1}$	$= 1/2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$	\dot{v}_2	$= \dot{v}_{-1} \times v_0 + v_{-1} \times \dot{v}_0$	$= 1 \times 5 + 2 \times 0$
\Downarrow v_3	$= \sin(v_0)$	$\sin(5)$	\Downarrow \dot{v}_3	$= \cos v_0 \times \dot{v}_0$	$= \cos(5) \times 0$
v_4	$= v_1 + v_2$	$= 0.6931 + 10$	\dot{v}_4	$= \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
v_5	$= v_4 - v_3$	$= 10.6931 + 0.9589$	\dot{v}_5	$= \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
y	$= v_5$	$= 11.6521$	$= \dot{y}$	\dot{v}_5	$= 5.5$

TABLE 2 – Reverse AD example, with $y = f(x_1, x_2) = \ln(x_1) + x_1x_2 - \sin(x_2)$ at $(x_1, x_2) = (2, 5)$. Setting $\bar{y} = 1$, $\partial y / \partial x_1$ and $\partial y / \partial x_2$ are computed in one reverse sweep.

Forward evaluation trace			Reverse adjoint trace		
v_{-1}	$= x_1$	$= 2$	\bar{x}_1	$= \bar{v}_{-1}$	$= 5.5$
v_0	$= x_2$	$= 5$	\bar{x}_2	$= \bar{v}_0$	$= 1.7163$
v_1	$= \ln(v_{-1})$	$= \ln(2)$	\bar{v}_{-1}	$= \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= 5.5$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$	\bar{v}_0	$= \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= 1.7163$
\Downarrow v_3	$= \sin(v_0)$	$= \sin(5)$	\Uparrow \bar{v}_{-1}	$= \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= 5$
v_4	$= v_1 + v_2$	$= 0.6931 + 10$	\bar{v}_0	$= \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= -0.2837$
v_5	$= v_4 - v_3$	$= 10.6931 + 0.9589$	\bar{v}_2	$= \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= 1$
y	$= v_5$	$= 11.6521$	\bar{v}_1	$= \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= 1$
			\bar{v}_3	$= \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= -1$
			\bar{v}_4	$= \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= 1$
			\bar{v}_5	$= \bar{y}$	$= 1$

Answer 3.

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.

Question 5 (6). Compute the *full*, *valid*, and *same* convolution (with kernel flipping) for the following 1D matrices : $[1, 2, 3, 4] * [1, 0, 2]$

Answer 5.

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.

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 $p = 1$, $d = 1$.

For each question, please fill up the following table :

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

Answer 7.