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

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. — $if\ x > 0 \lim_{\epsilon\to 0} \frac{g(x+\epsilon)-g(x)}{\epsilon} = \lim_{\epsilon\to 0} \frac{x+\epsilon-x}{\epsilon} = 1$
   — $if\ x < 0 \lim_{\epsilon\to 0} \frac{g(x+\epsilon)-g(x)}{\epsilon} = \lim_{\epsilon\to 0} \frac{0-0}{\epsilon} = 0$
   — $if\ x = 0$ we have two cases
     — $\lim_{\epsilon\to 0^-} \frac{g(x+\epsilon)-g(x)}{\epsilon} = \frac{0-0}{\epsilon} = 0$
     — $\lim_{\epsilon\to 0^+} \frac{g(x+\epsilon)-g(x)}{\epsilon} = \frac{x+\epsilon-x}{\epsilon} = 1$

     So the function is not differentiable at $x = 0$

   In summary

$$g'(x) = \begin{cases} 1, if\ x > 0 \\ 0, if\ x < 0 \end{cases}$$

2. The function $g(x)$ can be alternatively defined as :
   — $g(x) = xH(x)$
   — $g(x) = \int_{-\infty}^{\infty} H(t)\,dt$
3.

$$\lim_{k\to\infty} \sigma(x) = \frac{1}{1+e^{-kx}} = \begin{cases} 1, if\ x > 0 \\ 1/2, if\ x = 0 \\ 0, if\ x < 0 \end{cases} = H(x)$$

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. $S(\mathbf{x}+c)_i = \frac{e^{x_i+c}}{\sum_j e^{x_j+c}} = \frac{e^{x_i}e^c}{e^c\sum_j e^{x_j}} = S(\mathbf{x})$
2. $S(c\mathbf{x})_i = \frac{(e^{x_i})^c}{\sum_j (e^{x_j})^c} \neq S(\mathbf{x})_i$ except when $c=1$. This expression can be written as

$$S_c(\mathbf{x})_i = \frac{1}{\sum_j \left(\frac{e^{x_j}}{e^{x_i}}\right)^c}$$

so we can evaluate different cases :
   - if $c=0$, $S_c(\mathbf{x})_i = \frac{1}{n}$
   - if $\lim c \to \infty$ and
     - if $x_i > x_j$, $S_c(\mathbf{x})_i \to 1$
     - if $x_i < x_j$, $S_c(\mathbf{x})_i \to 0$
3. The component 1 of the vector is

$$S(\mathbf{x})_1 = \frac{e^{x_1}}{e^{x_1}+e^{x_2}} = \frac{1}{1+e^{-z}} = \sigma(z)$$

with $z \equiv x_1 - x_2$. In the same way the component 2 is

$$S(\mathbf{x})_2 = \frac{e^{x_2}}{e^{x_1}+e^{x_2}} = \frac{1}{1+e^{z}} = 1 - \frac{1}{1+e^{-z}} = 1-\sigma(z)$$

so

$$S(\mathbf{x}) = [\sigma(z), 1-\sigma(z)]$$

4. According to the property of invariance to translations, we can deduce that $S(\mathbf{x}) = S(\mathbf{x} - x_1)$, i.e $S([x_1, x_2, \ldots x_k]^T) = S([0, x_2 - x_1, x_3 - x_1, \ldots x_k - x_1]^T) = S([0, y_1, y_2, \ldots y_{k-1}])$

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. Considering the definition of $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and $\sigma(x) = \frac{1}{1+e^{-x}}$ we can obtain the relation $\sigma(x) = \frac{\tanh(x/2)+1}{2}$ replacing in the 2-neural layer network we arrive to

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

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		
	v_{-1}	$= x_1$	$= 2$
	v_0	$= x_2$	$= 5$
	v_1	$= \ln(v_1)$	$= \ln(2)$
	v_2	$= v_{-1} \times v_0$	$= 2 \times 5$
$\Downarrow$	v_3	$= \sin(v_0)$	$\sin(5)$
	v_4	$= v_1 + v_2$	$= 0.6931 + 10$
	v_5	$= v_4 - v_3$	$= 10.6931 + 0.9589$
	y	$= v_5$	$= 11.6521$

	Forward derivative trace		
	$= \dot{v}_{-1}$	$\dot{x}_1$	$= 1$
	$= \dot{v}_0$	$\dot{x}_2$	$= 0$
	$\dot{v}_1$	$= \dot{v}_{-1}/v_{-1}$	$= 1/2$
	$\dot{v}_2$	$= \dot{v}_{-1} \times v_0 + v_{-1} \times \dot{v}_0$	$= 1 \times 5 + 2 \times 0$
$\Downarrow$	$\dot{v}_3$	$= \cos v_0 \times \dot{v}_0$	$= \cos(5) \times 0$
	$\dot{v}_4$	$= \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
	$\dot{v}_5$	$= \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
	$= \dot{y}$	$\dot{v}_5$	$= 5.5$

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		
	v_{-1}	$= x_1$	$= 2$
	v_0	$= x_2$	$= 5$
	v_1	$= \ln(v_1)$	$= \ln(2)$
	v_2	$= v_{-1} \times v_0$	$= 2 \times 5$
$\Downarrow$	v_3	$= \sin(v_0)$	$= \sin(5)$
	v_4	$= v_1 + v_2$	$= 0.6931 + 10$
	v_5	$= v_4 - v_3$	$= 10.6931 + 0.9589$
	y	$= v_5$	$= 11.6521$

	Reverse adjoint trace		
	$\bar{x}_1$	$= \bar{v}_{-1}$	$= 5.5$
	$\bar{x}_2$	$= \bar{v}_0$	$= 1.7163$
	$\bar{v}_{-1}$	$= \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= 5.5$
	$\bar{v}_0$	$= \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= 1.7163$
$\Uparrow$	$\bar{v}_{-1}$	$= \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= 5$
	$\bar{v}_0$	$= \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= -0.2837$
	$\bar{v}_2$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= 1$
	$\bar{v}_1$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= 1$
	$\bar{v}_3$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= -1$
	$\bar{v}_4$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= 1$
	$\bar{v}_5$	$= \bar{y}$	$= 1$

with the definitions

$$\tilde{\omega_{kj}}^{(2)} \equiv \frac{\omega_{kj}^{(2)}}{2}; \;\; \tilde{\omega_{ji}}^{(1)} \equiv \frac{\omega_{ji}^{(1)}}{2}; \;\; \tilde{\omega_{k0}}^{(2)} \equiv \frac{1}{2} \sum_{j=1}^{M} \omega_{kj}^{(2)} + \omega_{k0}^{(2)}$$

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.

TABLE 3 – Forward AD exercise, with $y = f(x_1, x_2) = \frac{1}{x_1+x_2} + x_2^2 + \cos(x_1)$ 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	$= \frac{1}{v_{-1}+v_0}$	$= 0.1$
	v_2	$= v_0^2$	$= 36$
$\Downarrow$	v_3	$= v_1 + v_2$	$= 36.1$
	v_4	$= \cos(v_{-1})$	$= -0.99$
	v_5	$= v_4 + v_3$	$= 35.11$
	y	$= v_5$	$= 35.11$

	Forward derivative trace		
	$= \dot{v}_{-1}$	$\dot{x}_1$	$= 1$
	$= \dot{v}_0$	$\dot{x}_2$	$= 0$
	$\dot{v}_1$	$= -\frac{(\dot{v}_{-1}+\dot{v}_0)}{(v_{-1}+v_0)^2}$	$= -0.012$
	$\dot{v}_2$	$= 2v_0\dot{v}_0$	$= 0$
$\Downarrow$	$\dot{v}_3$	$= \dot{v}_1 + \dot{v}_2$	$= -0.012$
	$\dot{v}_4$	$= -\sin(v_{-1})\dot{v_{-1}}$	$= -0.14$
	$\dot{v}_5$	$= \dot{v}_4 + \dot{v}_3$	$= -0.153$
	$= \dot{y}$	$\dot{v}_5$	$= -0.153$

TABLE 4 – Reverse AD example, with $y = f(x_1, x_2) = \frac{1}{x_1+x_2} + x_2^2 + \cos(x_1)$ at $(x_1, x_2) = (3, 6)$. Setting $\bar{y} = 1$, $\partial y/\partial x_1$ and $\partial y/\partial x_2$ are computed in one reverse sweep.

	Forward evaluation trace		
	v_{-1}	$= x_1$	$= 3$
	v_0	$= x_2$	$= 6$
	v_1	$= \frac{1}{v_{-1}+v_0}$	$= 0.1$
	v_2	$= v_0^2$	$= 36$
$\Downarrow$	v_3	$= v_1 + v_2$	$= 36.1$
	v_4	$= \cos(v_{-1})$	$= -0.99$
	v_5	$= v_4 + v_3$	$= 35.11$
	y	$= v_5$	$= 35.11$

	Reverse adjoint trace		
	$\bar{x}_1$	$= \bar{v}_{-1}$	$= -0.153$
	$\bar{x}_2$	$= \bar{v}_0$	$= 11.98$
	$\bar{v}_{-1}$	$= \bar{v}_1\frac{\partial v_1}{\partial v_{-1}} + \bar{v}_4\frac{\partial v_4}{\partial v_{-1}}$	$= -0.153$
	$\bar{v}_0$	$= \bar{v}_1\frac{\partial v_1}{\partial v_0} + \bar{v}_2\frac{\partial v_2}{\partial v_0}$	$= 11.98$
$\Uparrow$	$\bar{v}_1$	$= \bar{v}_3\frac{\partial v_3}{\partial v_1}$	$= 1$
	$\bar{v}_2$	$= \bar{v}_3\frac{\partial v_3}{\partial v_2}$	$= 1$
	$\bar{v}_3$	$= \bar{v}_5\frac{\partial v_5}{\partial v_3}$	$= 1$
	$\bar{v}_4$	$= \bar{v}_5\frac{\partial v_5}{\partial v_4}$	$= 1$
	$\bar{v}_5$	$= \bar{y}$	$= 1$

Answer 4. Solutions are shown in Tab.3 and Tab.4

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

Answer 5. Considering the $N = 4$ elements of $[1, 2, 3, 4]$ and the $M = 3$ elements of $[1, 0, 2]$ the size of the full convolution is $N + (M - 1) = 6$. The elements in the full convolution are calculted as

$$\begin{bmatrix} [0,0,1,2,3,4] * [2,0,1,0,0,0] \\ [0,1,2,3,4] * [2,0,1,0,0] \\ [1,2,3,4] * [2,0,1,0] \\ [1,2,3,4] * [0,2,0,1] \\ [1,2,3,4,0] * [0,0,2,0,1] \\ [1,2,3,4,0,0] * [0,0,0,2,0,1] \end{bmatrix} = [1,2,5,8,6,8]$$

Therefore **Full** :$[1, 2, 5, 8, 6, 8]$. "Same" returns an output of length $\max[M, N] = 4$. and it is obtained from

$$\begin{bmatrix} [0,1,2,3,4] * [2,0,1,0,0] \\ [1,2,3,4] * [2,0,1,0] \\ [1,2,3,4] * [0,2,0,1] \\ [1,2,3,4,0] * [0,0,2,0,1] \end{bmatrix} = [2,5,8,6]$$

so **Same** :$[2, 5, 8, 6]$. Finally, "valid" returns an output of length $\max[M, N] - \min[M, N] + 1 = 4 - 3 + 1 = 2$ obtained using

$$\begin{bmatrix} [1, 2, 3, 4] * [2, 0, 1, 0] \\ [1, 2, 3, 4] * [0, 2, 0, 1] \end{bmatrix} = [5, 8]$$

Valid matrix is **Valid** :$[5, 8]$.

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. We use the expression $o = \frac{i-k+2p}{s} + 1$, where $o =$ output width size of the image, $i =$ input width size of the image, $k =$ size width of the kernel, $s =$ stride of the convolution operator and $p =$ padding. Using this expression in each layer we obtain
   — First layer : $o = \left(\frac{256-8+0}{2} + 1\right) \times \left(\frac{256-8+0}{2} + 1\right) \times 64 = 125 \times 125 \times 64$
   — Second layer : $o = \left(\frac{125-5+0}{5} + 1\right) \times \left(\frac{125-5+0}{5} + 1\right) \times 64 = 25 \times 25 \times 64$
   — Third layer : $o = \left(\frac{25-4+2}{1} + 1\right) \times \left(\frac{25-4+2}{1} + 1\right) \times 128 = 24 \times 24 \times 128$

   thus the third layer has $24 * 24 * 128 = 73728$ dimensions.
2. The number of parameters p is defined by $p = k^2 * c * N + B_c$ where $N =$ is the number of kernels, $c =$ the number of channels of the input image and B_c the number of biases, the other variables have the same definition as above. In the third layer $p = 4^2 * 64 * 128 = 131072$

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. In this case we use the expression $o = \frac{i-k'+2p}{s} + 1$, with $k' = d(k-1) + 1$, we observe that when $d = 1$ (no dilation), $k' = k$ as used in the last exercise. Filling the table we obtain :

		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	5	1	4
	(c)	8	1	1	4	2	4