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. (a) For $x > 0$ we can write :
   $\lim_{\epsilon \to 0} \frac{g(x+\epsilon)-g(x)}{\epsilon} \xrightarrow{g(x+\epsilon)=x+\epsilon, \quad g(x)=x} \lim_{\epsilon \to 0} \frac{x+\epsilon)-x}{\epsilon} = \frac{\epsilon}{\epsilon} = 1$
   (b) For $x < 0$ we can write :
   $\lim_{\epsilon \to 0} \frac{g(x+\epsilon)-g(x)}{\epsilon} \xrightarrow{g(x+\epsilon)=g(x)=0} \lim_{\epsilon \to 0} \frac{0-0}{\epsilon} = 1$
   (c) For $x = 0$ we can write :
   $\frac{d}{dx}g(x) = \lim_{\epsilon \to 0^-} \frac{g(x+\epsilon)-g(x)}{\epsilon} = \lim_{\epsilon \to 0^-} \frac{g(0+\epsilon)-g(0)}{\epsilon} = \lim_{\epsilon \to 0^-} \frac{0-0}{\epsilon} = 0$

   $\frac{d}{dx}g(x) = \lim_{\epsilon \to 0^+} \frac{g(x+\epsilon)-g(x)}{\epsilon} = \lim_{\epsilon \to 0^+} \frac{g(0+\epsilon)-g(0)}{\epsilon} = \lim_{\epsilon \to 0^+} \frac{\epsilon-0}{\epsilon} = 1$
   Using the above equations we can see that g(x) is not differentiable at point $x = 0$.

2. Here are the two equivalent functions using Heaviside step function :
   (a) $g(x) = \int_{-\infty}^{x} H(x)dx$
   (b) $g(x) = x * H(x)$

3. In order to show that $H(x)$ can be approximated by sigmoid we need to show that it can approximate any part of it.
   (a) Lets solve it for large k and $x > 0$:
   $\lim_{+\infty} \sigma(x) = \lim_{+\infty} \frac{1}{1+e^{-kx}} = \lim_{+\infty} \frac{1}{1+e^{-\infty}} = 1$

(b) For large k and $x = 0$ we have :
$\lim_{+\infty} \sigma(x) = \lim_{+\infty} \frac{1}{1+e^{-kx}} = \lim_{+\infty} \frac{1}{1+e^{0}} = \frac{1}{2}$

(c) For large k and $x < 0$ we have :
$\lim_{+\infty} \sigma(x) = \lim_{+\infty} \frac{1}{1+e^{-kx}} = \lim_{+\infty} \frac{1}{1+e^{+\infty}} = 0$

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. Lets use the definition of softmax function to show that it is translation-invariant :
$S(\boldsymbol{x}+c)_i = \frac{e^{\boldsymbol{x}_i + c}}{\sum_j e^{\boldsymbol{x}_j + c}} = \frac{e^{\boldsymbol{x}_i} e^{c}}{e^{c} \sum_j e^{\boldsymbol{x}_j}} = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}} = S(\boldsymbol{x})_i \quad \xrightarrow{c-is-scalar} \quad S(\boldsymbol{x}+c) = S(\boldsymbol{x})$

2. If $c = 1$, invariant under scalar multiplication will be equal to original softmax.
If we take $c = 0$ the problem is that softmax will have a uniform output. See the following equations :
$S_c(\boldsymbol{x})_i \xrightarrow{c=0} \frac{e^{0 * \boldsymbol{x}_i}}{\sum_j e^{0 * \boldsymbol{x}_j}} = \frac{1}{\sum_j 1} = \frac{1}{Size-of-vector-X}$

   To prove that softmax is not invariant under scalar multiplication we need to find a counterexample, take $c = 0$ or $c = +\infty$ for instance : $S_c(\boldsymbol{x})_i = \frac{e^{c\boldsymbol{x}_i}}{\sum_j e^{c\boldsymbol{x}_j}} = \frac{e^{\boldsymbol{x}_i c}}{\sum_j (e^{\boldsymbol{x}_j})^c} \neq S(\boldsymbol{x})_i$

   If we take $c = +\infty$, softmax will behave like argmax which means it will be 1 only for the highest value and 0 for all other values.

3. X is a 2-dimensional array so it can be written as $X_2 d = [x_1, x_2]$, So we can calculate softmax for x_1 and x_2 :
$S(x_1) = \frac{e^{\boldsymbol{x}_1}}{e^{\boldsymbol{x}_1} + e^{\boldsymbol{x}_2}} \xrightarrow{divide-by-x_1} \frac{1}{1+(e^{\boldsymbol{x}_2 - \boldsymbol{x}_1})} = \frac{1}{1+(e^{-(\boldsymbol{x}_1 - \boldsymbol{x}_2)})} = \sigma x_1, x_2$
$S(x_2) = \frac{e^{\boldsymbol{x}_2}}{e^{\boldsymbol{x}_1} + e^{\boldsymbol{x}_2}} \xrightarrow{divide-by-x_2} \frac{1}{1+(e^{\boldsymbol{x}_1 - \boldsymbol{x}_2})} = \frac{1+e^{\boldsymbol{x}_1 - \boldsymbol{x}_2} - e^{\boldsymbol{x}_1 - \boldsymbol{x}_2}}{1+(e^{\boldsymbol{x}_1 - \boldsymbol{x}_2})} = 1 - \frac{e^{\boldsymbol{x}_1 - \boldsymbol{x}_2}}{1+(e^{\boldsymbol{x}_1 - \boldsymbol{x}_2})} = 1 - \sigma(x_1 - x_2)$
Then $S(X_2 d)$ can be written as $[\sigma(x_1 - x_2), 1 - \sigma(x_1 - x_2)]$.

4. Let $X = [x_1, x_2, ..., x_K]$, since softmax is translation-invariant we can represent a K-dimensional vector using $K-1$ parameters by subtracting one of the elements from all of the elements. By subtracting x_1 from all of the elements we have $S(X) = S(X - x_1) = S([x_1 - x_1, x_2 - x_1, ..., x_K - x_1]) = S([0, x_2 - x_1, ..., x_K - x_1])$ which contains $K-1$ different elements.

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 know $\sigma(x) = \frac{1}{1+e^{-x}}$, and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1} = 2\sigma(2x) + 1$. We can conclude that $\sigma(x) = \frac{1}{2}\tanh(\frac{x}{2} + 1)$.

Lets apply it in the equation :

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

Until now, we show that there exists an equivalent network such that $y(x, \Theta', \tanh) = y(x, \Theta, \sigma)$ for all $x \in \mathbb{R}^D$.
Now we should express that Θ' is a function of Θ : $\tilde{\omega_{k0}^{(2)}} = (\sum_{j=1}^{M} \frac{\omega_{kj}^{(2)}}{2} + \omega_{k0}^{(2)})$, $\tilde{\omega_{kj}^{(2)}} = \frac{\omega_{kj}^{(2)}}{2}$,
and $\tilde{\omega_{ji}^{(1)}} = \frac{\omega_{ji}^{(1)}}{2}$.

TABLE 1 – Forward AD example, with $y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ 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	$= v_{-1} \times v_0$	$= 2 \times 5$
⇓ 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		
$= \dot{v}_{-1}$	$\dot{x}_1$	$= 1$
$= \dot{v}_0$	$\dot{x}_2$	$= 0$
$\dot{v}_1$	$= \dot{v}_{-1}/v_{-1}$	$= 1/2$
$\dot{v}_2$	$= \dot{v}_{-1} \times v_0 + v_{-1} \times \dot{v}_0$	$= 1 \times 5 + 2 \times 0$
⇓ $\dot{v}_3$	$= \cos v_0 \times \dot{v}_0$	$= \cos(5) \times 0$
$\dot{v}_4$	$= \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
$\dot{v}_5$	$= \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
$= \dot{y}$	$\dot{v}_5$	$= 5.5$

TABLE 2 – Reverse AD example, with $y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ 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	$= v_{-1} \times v_0$	$= 2 \times 5$
⇓ 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		
$\bar{x}_1$	$= \bar{v}_{-1}$	$= 5.5$
$\bar{x}_2$	$= \bar{v}_0$	$= 1.7163$
$\bar{v}_{-1}$	$= \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= 5.5$
$\bar{v}_0$	$= \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= 1.7163$
⇑ $\bar{v}_{-1}$	$= \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= 5$
$\bar{v}_0$	$= \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= -0.2837$
$\bar{v}_2$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= 1$
$\bar{v}_1$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= 1$
$\bar{v}_3$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= -1$
$\bar{v}_4$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= 1$
$\bar{v}_5$	$= \bar{y}$	$= 1$

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 4, 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.

TABLE 3 – 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	$= v_{-1} + v_0$	$= 3 + 6$
	v_2	$= v_0 \times v_0$	$= 6 \times 6$
$\Downarrow$	v_3	$= 1/v_1$	$= \frac{1}{9}$
	v_4	$= \cos(v_{-1})$	$= \cos(3)$
	v_5	$= v_3 + v_2$	$= \frac{1}{9} + 36$
	v_6	$= v_5 + v_4$	$= 36.0278 - 0.9899$
	y	$= v_6$	$= 35.0379$

	Forward derivative trace		
	$= \dot{v}_{-1}$	$\dot{x}_1$	$= 1$
	$= \dot{v}_0$	$\dot{x}_2$	$= 0$
	$\dot{v}_1$	$= \dot{v}_{-1} + \dot{v}_0$	$= 1$
	$\dot{v}_2$	$= 2 \times \dot{v}_0 \times v_0$	$= 2 \times 0 \times 6$
$\Downarrow$	$\dot{v}_3$	$= -\dot{v}_1/v_1 \times v_1$	$= -1/81$
	$\dot{v}_4$	$= -\sin(v_{-1}) \times \dot{v_{-1}}$	$= -\sin(3)$
	$\dot{v}_5$	$= \dot{v}_3 + \dot{v}_2$	$= -0.01234 + 0$
	$\dot{v}_6$	$= \dot{v}_5 + \dot{v}_4$	$= -0.0123 - 0.1411$
	$= \dot{y}$	$\dot{v}_6$	$= -0.1534$

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

Answer 5. Full : From page 344 of Deep Learning book we know that "we should add enough zeros to have an output of width $m + k - 1$" which is equal to $4 + 3 - 1 = 6$ for this problem. Now that we know the size of output we can calculate needed padding by using the following formula :
$out = \lfloor \frac{in-k+2p}{s} \rfloor + 1 \xrightarrow{s=1,p=?} p = 2$
$\big[0, 0, 1, 2, 3, 4, 0, 0\big] * \big[2, 0, 1\big] \xrightarrow{convolve}$
$$\begin{bmatrix} 0 \cdot 2 + 0 \cdot 0 + 1 \cdot 1, 0 \cdot 2 + 1 \cdot 0 + 2 \cdot 1, 1 \cdot 2 + 2 \cdot 0 + 3 \cdot 1, 2 \cdot 2 + 3 \cdot 0 + 4 \cdot 1, \\ 3 \cdot 2 + 4 \cdot 0 + 0 \cdot 1, 4 \cdot 2 + 0 \cdot 0 + 0 \cdot 1 \end{bmatrix} = \big[1, 2, 5, 8, 6, 8\big] ;$$

Valid : It means that we should use no padding method, and only convolve the flipped kernel with stride 1. Lets do the math :
$\big[1, 2, 3, 4\big] * \big[2, 0, 1\big] \xrightarrow{convolve} \big[1 \cdot 2 + 2 \cdot 0 + 3 \cdot 1, 2 \cdot 2 + 3 \cdot 0 + 4 \cdot 1\big] = \big[5, 8\big] ;$
Same : It means that the output size should be the same as input. Using the above mentioned formula, we can calculate the padding size. Which means :
$4 = \lfloor \frac{4-3+2p}{1} \rfloor + 1 \xrightarrow{s=1,p=?} p = 1.$
Lets compute the final result of convolution : $\big[0, 1, 2, 3, 4, 0\big] * \big[2, 0, 1\big] =$
$$\begin{bmatrix} 0 \cdot 2 + 1 \cdot 0 + 2 \cdot 1, 1 \cdot 2 + 2 \cdot 0 + 3 \cdot 1, 2 \cdot 2 + 3 \cdot 0 + 4 \cdot 1, \\ 3 \cdot 2 + 4 \cdot 0 + 0 \cdot 1 \end{bmatrix} = \big[2, 5, 8, 6\big].$$

TABLE 4 – 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	$= v_0 \times v_0$	$= 6 \times 6$
$\Downarrow$	v_3	$= 1/v_1$	$= \frac{1}{9}$
	v_4	$= \cos(v_{-1})$	$= \cos(3)$
	v_5	$= v_3 + v_2$	$= \frac{1}{9} + 36$
	v_6	$= v_5 + v_4$	$= 36.0278 - 0.9899$
	y	$= v_6$	$= 35.0379$

	Reverse adjoint trace		
	$\bar{x}_1$	$= \bar{v}_{-1}$	$= -0.1534$
	$\bar{x}_2$	$= \bar{v}_0$	$= 11.9877$
	$\bar{v}_0$	$= \bar{v}_0 + \bar{v}_1 \frac{\partial v_1}{\partial v_0}$	$= 12 - 0.0123$
	$\bar{v}_{-1}$	$= \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= -0.0123 - 0.1411$
$\Uparrow$	$\bar{v}_0$	$= \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= 12$
	$\bar{v}_1$	$= \bar{v}_3 \frac{\partial v_3}{\partial v_1}$	$= -1/81$
	$\bar{v}_{-1}$	$= \bar{v}_4 \frac{\partial v_4}{\partial v_{-1}}$	$= -0.1411$
	$\bar{v}_2$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_2}$	$= 1$
	$\bar{v}_3$	$= \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= 1$
	$\bar{v}_4$	$= \bar{v}_6 \frac{\partial v_6}{\partial v_4}$	$= 1$
	$\bar{v}_5$	$= \bar{v}_6 \frac{\partial v_6}{\partial v_5}$	$= 1$
	$\bar{v}_6$	$= \bar{y}$	$= 1$

Question 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. First we need to calculate dimensionality of the output of previous layers one by one in order to find the output of the last layer. We can calculate the output dimensionality of each layer using input size (in), kernel size (k), stride size (s), and the amount of zero padding (p) on the borders using the following formula : $out = \lfloor \frac{in-k+2p}{s} \rfloor + 1$
   Lets start with calculating the output size of first layer :
   $256 * 256 * 3 \xrightarrow{64,8\times8,s=2,p=0} (\lfloor \frac{256-8+2*0}{2} \rfloor + 1) * (\lfloor \frac{256-8+2*0}{2} \rfloor + 1) * 64 = 125 * 125 * 64$ Now that we have the output size of first layer we can use it as an input size of the second layer :
   $125 * 125 * 64 \xrightarrow{maxpooling,5\times5,s=5,p=0} (\lfloor \frac{125-5+2*0}{5} \rfloor + 1) * (\lfloor \frac{125-5+2*0}{5} \rfloor + 1) * 64 = 25 * 25 * 64$
   Finally, we can calculate the dimensionality of the output of the last layer :
   $25 * 25 * 64 \xrightarrow{128,4\times4,s=1,p=1} (\lfloor \frac{25-4+2*1}{1} \rfloor + 1) * (\lfloor \frac{25-4+2*1}{1} \rfloor + 1) * 128 = 24 * 24 * 128$
2. In last layer we convolved 128 4×4 kernels with the output of previous layer which contains 64 channels. As a result, the number of parameters needed for the last layer is $64*128*4*4 = 131072$.

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,

		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	3	2	4
	(c)	8	1	1	4	2	4

From previous question we know that output size can be calculated using the following formula :

$$out = \lfloor \frac{in - (d(k-1)+1) + 2p}{s} \rfloor + 1$$

1. The output shape (o) of the first layer is $(64, 32, 32)$.
   (a) $k = 8$, $d = 1$, $in = 64 \rightarrow 32 = \lfloor \frac{64-8+2p}{s} \rfloor + 1$ one possible solution is $p = 3$, $s = 2$.
   (b) $d = 7$, $s = 2$, $in = 64 \rightarrow 32 = \lfloor \frac{64-(7(k-1)+1)+2p}{2} \rfloor + 1 \rightarrow 0 = \lfloor \frac{-7(k-1)+2p}{2} \rfloor$ one possible solution is $p = 3$, $k = 2$.
2. The output shape of the second layer is $(64, 8, 8)$. Assume $p = 0$ and $d = 1$.
   (a) $p = 0$, $d = 1$, $in = 32$, max pooling, non-overlapping window $\xrightarrow{k=s} 8 = \lfloor \frac{32-(1(k-1)+1)}{k} \rfloor + 1$, one possible solution is $k = s = 4$.
   (b) $p = 0$, $d = 1$, $in = 32$, $k = 8$, $s = 4$ max pooling $\rightarrow$ out $= \lfloor \frac{32-(1(8-1)+1)}{4} \rfloor + 1$, output shape will be $(64, 7, 7)$.
3. The output shape of the last layer is $(128, 4, 4)$.
   (a) $d = 1$, $p = 0$, $in = 8 \rightarrow 4 = \lfloor \frac{8-k}{s} \rfloor + 1 \rightarrow$ one possible solution is $k = 2$, $s = 2$.
   (b) $d = 2$, $p = 2 \rightarrow 4 = \lfloor \frac{8-(2(k-1)+1)+4}{s} \rfloor + 1 \rightarrow$ possible solutions are $k = 3$, $s = 2$ or $k = 5$, $s = 1$.
   (c) Assume $p = 1$, $d = 1 \rightarrow 4 = \lfloor \frac{8-k+2}{s} \rfloor + 1 \rightarrow$ possible solutions are $k = 4$, $s = 2$ or $k = 7$, $s = 1$.

Some parts had different answers so I wrote more than one.